

# PR #56844 完整报告

vllm-project/vllm

[Bugfix] Validate routed-expert prompt offsets before engine submission

合并时间: 2026-09-15 09:17

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56844>

## 执行摘要

- 一句话: 提交前校验 routed-experts 偏移, 避免 EngineCore 崩溃
- 推荐动作: 值得精读 (篇幅小、信息密度高)。关注三个设计决策: (a) 校验位置选在 InputProcessor 而非 SamplingParams 或调度器, 理由是覆盖 OpenAI 与非 OpenAI 全部生成调用且早于 EngineCore 提交; (b) 采用 '拒绝' 而不是 'clamp', 避免把非法请求静默改写为语义不明的合法请求; (c) 闭区间语义——把 `offset == prompt_len` 视为合法的空切片而非错误, 并同步放宽调度器断言。建议读者顺手排查自己代码里是否存在绕过 InputProcessor 直接投递 Request 的路径, 以及在 python -O 下调度器 assert 失效后的行为。若计划进一步加固, 可考虑把下界检查下沉到 SamplingParams 校验或让调度器显式抛错替代 assert。

## 功能与动机

Issue #55750 报告: 带 routed-expert 元数据的请求会导致 EngineCore 崩溃, 并让后续请求一起失败 (服务级可用性损失)。#55764 与 #55765 分别提交了重叠修复, 本 PR 按 submitter 要求把两者合并到一处: PR body 明确 'Invalid offsets become validation errors instead of reaching the engine or overflowing MessagePack serialization', 且 'An offset equal to the prompt length is valid and returns an empty prompt routing slice'。该条目在 #51891 (Patch the Planet 安全加固, GHSA-98fv-r3w4-mrrv) 中对应 PTP-VLLM-111 'routed-expert prompt offset', 本 PR 只取其中的偏移校验切片。

## 实现拆解

1. 入口校验 (vllm/v1/engine/input\_processor.py, InputProcessor.process\_inputs): 在 `isinstance(params, SamplingParams)` 分支中先算出 `prompt_len = length_from_prompt_token_ids_or_embeds(prompt_token_ids, prompt_embeds)`, 随后用一行 `if not 0 <= sampling_params.routed_experts_prompt_start <= prompt_len:` 抛出 `VLLMValidationError`, 携带 `parameter` 与 `value` 便于前端映射成 400 级错误。选择这一层是因为它是 OpenAI 与非 OpenAI 生成调用共用的提交边界, 非法值在此被拦下就不会到达 EngineCore。
2. 复用长度计算 (同文件): 原先 `max_tokens` 推导和 `trace_decode_token_ids` 归一化各自调用一次 `length_from_prompt_token_ids_or_embeds`, 现在统一用局部变量 `prompt_len`, 既省掉重复求值, 也保证校验上界与实际使用的 prompt 长度口径一致。未引入新 helper, 未做 clamp——非法偏移直接被拒绝, 而不是被静默改写。

3. 放宽调度器边界断言 (`vllm/v1/core/sched/scheduler.py`, `Scheduler.update_from_output` 的 `prefill` 完成分支) : 把 `assert prompt_start < request.num_prompt_tokens` 改为 `assert prompt_start <= request.num_prompt_tokens`, 使 `offset` 等于 `prompt` 长度时继续走 `routed_experts_mgr.get(block_ids, num_prompt_tokens, token_start=prompt_start)` 并返回空切片, 与入口校验的闭区间语义对齐。
4. 测试配套 (`tests/test_request_input_bounds.py`、`tests/v1/core/test_scheduler.py`) : 前者新增参数化用例 `test_routed_experts_offset_validated_before_engine_submission`, 覆盖 `-1, 0, 2, 3, 4, 2**127` 六个偏移, 用 `Mock(spec=InputProcessor)` 伪造 `self` 直接在 CPU 上调用未绑定方法, 验证即使 `routed-expert` 输出关闭也照样拒绝; 后者新增 `test_routed_experts_prompt_start_at_prompt_end`, 断言 `offset == num_prompt_tokens` 时 `manager.get` 被以 `token_start=num_prompt_tokens` 调用且输出形状为 `(0, 1, 1)`。移除校验与边界修复后 4 个用例失败, 证明测试确实能捕获两个缺陷。
5. 无配置 / 部署改动: 未新增开关或环境变量, 属于纯粹的请求边界行为收紧; 作者明确说明未做 GPU / 模型评估, 并声明不修复 CI 重试中另一个无关的 ROCm GPU hang。

关键文件:

- `vllm/v1/engine/input_processor.py` (模块 输入处理; 类别 `source`; 类型 `core-logic`; 符号 `InputProcessor.process_inputs, length_from_prompt_token_ids_or_embeds, _normalize_trace_replay_params`) : 唯一的校验落点: 在请求提交 `EngineCore` 之前对 `routed_experts_prompt_start` 做闭区间检查并抛 `VLLMValidationError`, 同时把 `prompt` 长度计算收敛为一个局部变量供 `max_tokens` 与 `trace-replay` 复用。
- `vllm/v1/core/sched/scheduler.py` (模块 调度器; 类别 `source`; 类型 `core-logic`; 符号 `Scheduler.update_from_output`) : 调度器侧边界语义同步: 把 `prompt_start` 断言从 `'<'` 放宽为 `'<='`, 使 `offset` 等于 `prompt` 长度时返回空路由切片而不是断言崩溃。
- `tests/test_request_input_bounds.py` (模块 输入校验测试; 类别 `test`; 类型 `test-coverage`; 符号 `test_routed_experts_offset_validated_before_engine_submission`) : 参数化回归用例覆盖 `-1 / 0 / 2 / 3 / 4 / 2**127` 六种偏移, 用 `Mock(spec=InputProcessor)` 在 CPU 上直接驱动未绑定方法, 验证即使 `routed-expert` 输出关闭也必须在提交前拒绝非法偏移。
- `tests/v1/core/test_scheduler.py` (模块 调度器测试; 类别 `test`; 类型 `test-coverage`; 符号 `test_routed_experts_prompt_start_at_prompt_end`) : 新增 `test_routed_experts_prompt_start_at_prompt_end`, 用 `Mock` 的路由管理器断言 `offset == num_prompt_tokens` 时返回空切片形状 `(0, 1, 1)`, 锁定边界语义。

关键符号: `InputProcessor.process_inputs`, `Scheduler.update_from_output`, `length_from_prompt_token_ids_or_embeds`, `InputProcessor._normalize_trace_replay_params`, `test_routed_experts_offset_validated_before_engine_submission`, `test_routed_experts_prompt_start_at_prompt_end`

## 关键源码片段

`vllm/v1/engine/input_processor.py`

唯一的校验落点：在请求提交 EngineCore 之前对 routed\_experts\_prompt\_start 做闭区间检查并抛 VLLMValidationError，同时把 prompt 长度计算收敛为一个局部变量供 max\_tokens 与 trace-replay 复用。

```
# vllm/v1/engine/input_processor.py —— InputProcessor.process_inputs 中 SamplingParams 分支
```

```
if isinstance(params, SamplingParams):
```

```
    # TODO: can we avoid cloning here in multiproc case?
```

```
    sampling_params = params.clone()
```

```
    # 只计算一次 prompt 长度：既作为路由偏移校验的上界，
```

```
    # 也复用于 max_tokens 推导与 trace-replay 归一化，
```

```
    # 避免同一长度在多处各算一遍导致口径漂移
```

```
    prompt_len = length_from_prompt_token_ids_or_embeds(
        prompt_token_ids, prompt_embeds
    )
```

```
    # 提交 EngineCore 之前的入口校验：闭区间，offset == prompt_len 属于合法边界
```

```
    # （下游返回空路由切片）；负数会被 MessagePack 序列化放大，
```

```
    # 超大值（如 2**127）同样必须在这里拒绝，否则会打挂引擎
```

```
    if not 0 <= sampling_params.routed_experts_prompt_start <= prompt_len:
```

```
        raise VLLMValidationError(
            f"routed_experts_prompt_start must be between 0 and "
            f"the prompt length ({prompt_len}), inclusive.",
            parameter="routed_experts_prompt_start",
            value=sampling_params.routed_experts_prompt_start,
        )
```

```
    # 复用同一个 prompt_len，保证 max_tokens 与上面校验使用完全相同的长度口径
```

```
    if sampling_params.max_tokens is None:
```

```
        sampling_params.max_tokens = (
            self.model_config.max_model_len - prompt_len
        )
```

```
    sampling_params.update_from_generation_config(
```

```
        self.generation_config_fields,
        self.renderer.get_eos_token_id(),
    )
```

```
    if self.tokenizer is not None:
```

```
        sampling_params.update_from_tokenizer(self.tokenizer)
```

```
    if sampling_params.trace_decode_token_ids:
```

```
        # 原先此处会再调用一次 length_from_prompt_token_ids_or_embeds
        self._normalize_trace_replay_params(sampling_params, prompt_len)
```

```
    else:
```

```
        pooling_params = params.clone()
```

## vllm/v1/core/sched/scheduler.py

调度器侧边界语义同步：把 prompt\_start 断言从 '<' 放宽为 '<='，使 offset 等于 prompt 长度时返回空路由切片而不是断言崩溃。

```

# vllm/v1/core/sched/scheduler.py —— Scheduler.update_from_output 的 prefill 完成分支
routed_experts = None
if (
    self.enable_return_routed_experts
    and routing_data is not None
    and new_token_ids
):
    req_offset = routing_offsets[req_id]
    end = req_offset + num_tokens_scheduled
    block_ids = self._re_block_ids.pop(req_id, [])
    if num_output_tokens_before == 0:
        # Prefill 完成：按调度时快照的 block ID 读取整段 prompt 路由，
        # 快照方式可免受异步抢占影响
        if (
            request.sampling_params is not None
            and request.sampling_params.routed_experts_prompt_start is not None
        ):
            prompt_start = (
                request.sampling_params.routed_experts_prompt_start
            )
            # 上界由 < 放宽为 <=: offset 恰好等于 prompt 长度是合法输入，
            # 此时 get() 返回形状为 (0, 1, 1) 的空切片，
            # 而不是触发 AssertionError 把 EngineCore 打挂
            assert prompt_start <= request.num_prompt_tokens
        else:
            prompt_start = 0
        # 注意：这里用的上界是 request.num_prompt_tokens，
        # 与 InputProcessor 侧的 prompt_len 是两个来源，需保持口径一致
        routed_experts = self.routed_experts_mgr.get(
            block_ids,
            request.num_prompt_tokens,
            token_start=prompt_start,
        )

```

## tests/test\_request\_input\_bounds.py

参数化回归用例覆盖 -1 / 0 / 2 / 3 / 4 / 2\*\*127 六种偏移，用 Mock(spec=InputProcessor) 在 CPU 上直接驱动未绑定方法，验证即使 routed-expert 输出关闭也必须在提交前拒绝非法偏移。

```

# tests/test_request_input_bounds.py
@pytest.mark.parametrize("offset", [-1, 0, 2, 3, 4, 2**127])
def test_routed_experts_offset_validated_before_engine_submission(offset):
    """Reject invalid offsets even when routed-expert output is disabled."""
    # 用 Mock(spec=InputProcessor) 伪造 self，再直接调用未绑定方法，
    # 这样无需真实引擎即可在 CPU 上覆盖提交路径；
    # 2**127 用来验证超大值同样在校验层被拒绝
    processor = Mock(spec=InputProcessor)
    processor.tokenizer = None
    processor.generation_config_fields = {}

```

```

processor.renderer = Mock()
processor.renderer.get_eos_token_id.return_value = None
processor.vllm_config = Mock()
params = SamplingParams(routed_experts_prompt_start=offset)
prompt = {"type": "token", "prompt_token_ids": [1, 2, 3]}

if not 0 <= offset <= 3:
    # 非法偏移必须抛出带参数名的校验错误，参数名可被前端映射为具体字段提示
    with pytest.raises(VLLMValidationError, match="routed_experts_prompt_start"):
        InputProcessor.process_inputs(
            processor, "invalid", prompt, params, ("generate",)
        )
    params.routed_experts_prompt_start = 0

# 合法偏移（含边界值 3）应原样透传到 Request.sampling_params
request = InputProcessor.process_inputs(
    processor, "valid", prompt, params, ("generate",)
)
assert (
    request.sampling_params.routed_experts_prompt_start
    == params.routed_experts_prompt_start
)

```

## 评论区精华

本 PR 几乎没有技术层面的行内评审交锋：`review_comments_count` 为 0，仅收到 `claude[bot]` 的 'This pull request is from a fork — automated review is disabled' 提示与 `jeejeelee` 的一句 APPROVED。作者随后在 issue 区留言 '@AndreasKaratzas Can I get a stamp?'，说明合并前曾主动寻求第二个批准。真正承载设计权衡的是 PR body 本身：(1) 明确合并 #55764/#55765 并给出署名顺序（@bhushanasati25 主作者、@meghaagr13 次作者含 prompt-end 回归测试），并声明 'attribution does not imply approval of this consolidation by the original authors'；(2) 解释为何 'the range check lives in one place, with no extra helper or scheduler clamping'，即拒绝 clamp 方案；(3) 划定范围，只做 prompt offset，不含 #51891 的其余校验。issue 评论则集中在 `/ci run`、`/ci run all`、`/ci cancel`、重跑等 CI 操作以及 @AndreasKaratzas 的催审，无技术讨论。

- 合并重叠修复 #55764 / #55765 并确定署名与范围 (design): 采用单点校验：范围检查只放 InputProcessor，不引入额外 helper，也不在调度器做 clamp；核对后确认 consolidation 覆盖了两个原始补丁的能力。
- 校验落点与实现方式的取舍（拒绝 vs clamp） (design): 确定在提交边界拒绝非法偏移，而非静默改写为合法值；不新增抽象层。
- 范围边界：不覆盖 #51891 的其余校验项 (design): 范围收敛为 PTP-VLLM-111 单项；其余项仍由 #51891 承接。
- 评审与合并流程 (other): 最终以 1 个人工批准 + CI 通过合并，逐行人工 review 的记录缺失。

## 风险与影响

- 风险:

1. 校验覆盖面存在缺口：校验位于 `InputProcessor.process_inputs` 的 `SamplingParams` 分支，任何绕过该入口直接构造 `Request` 送入 `EngineCore` 的内部调用路径（自定义 `Engine` 集成、内部工具、测试夹具）仍可带非法偏移进入调度器，而调度器侧只有 `assert`——在 `python -O` 下会被整体剥离，且它不检查下界。也就是说 `engine` 端仍是 '崩溃' 而非 '拒绝'。
2. `prompt` 长度口径不一致风险：入口用 `length_from_prompt_token_ids_or_embeds(prompt_token_ids, prompt_embeds)`，调度器用 `request.num_prompt_tokens`。若在多模态占位、`encoder-decoder` 拆分或截断类参数下两者出现偏差，会出现校验通过但 `assert prompt_start <= request.num_prompt_tokens` 触发的情况，现有测试未覆盖这种跨口径场景。
3. 边界语义变更（回归面）：`offset == prompt_len` 由 'assert 失败' 变为返回形状 `(0, 1, 1)` 的空 `numpy` 数组。下游 `detokenizer` 与 `OpenAI` 响应序列化对空 `routed_experts`（而非缺字段）的处理未经端到端测试验证，客户端若假设该字段非空可能存在兼容性问题。
4. 异常类型变化：非法偏移现在抛 `VLLMValidationError`，而在当前 `main` 上它已不再是 `ValueError` 子类（#51891 提到相应测试需要更新）。依赖捕获 `ValueError` 的上层调用方需要相应调整。
5. 评审强度不足：fork PR 的自动化评审被禁用，仅 1 个人工 APPROVED，作者自述 'Submitted as a draft pending the human submitter's line-by-line review'，实际合并时并无逐行 review 记录，属于过程性风险。
6. 性能：影响可忽略——校验为  $O(1)$ ，且把两次 `length_from_prompt_token_ids_or_embeds` 合并为一次，属于轻微正向变化。
  - 影响：用户侧：非法 `routed_experts_prompt_start`（负数、超过 `prompt` 长度、 $2^{**127}$  这类极值）从 '打挂 `EngineCore` 并使 `/health` 返回 503、后续请求全部失败' 降级为单个请求的校验失败；合法请求路径与生成结果完全不变，`offset == prompt_len` 由崩溃变为返回空路由数据。系统侧：在请求提交边界新增一个闭区间检查并复用已有的 `prompt` 长度计算，对吞吐与显存无实质影响，无配置或部署变更。团队侧：合并了 #55764 与 #55765 两个重叠修复，收敛了重复评审成本，并把 '边界校验放 `InputProcessor`、拒绝而非 `clamp`、不引入 `helper`' 确立为后续 #51891 其它 PTP 项的落地范式。
  - 风险标记：请求校验边界变更，绕过 `InputProcessor` 的路径仍无防护，调度器侧仅有 `assert`（`-O` 下失效），`prompt` 长度两处口径可能不一致，空路由切片下游兼容性未端到端验证，人工评审覆盖较少

## 关联脉络

- PR #55764 [Bugfix] Avoid routed-expert metadata crash at prompt boundary: 同一 issue #55750 的原始修复之一，改动文件与本 PR 高度重合（`input_processor.py` / `scheduler.py` / 同名测试），作者 @bhushanasati25 亦为本 PR 首个 commit 的作者；本 PR 是其集中化版本。

- PR #55765 Fix for negative prompt length: 第二个重叠修复（已 closed、标记 needs-rebase），同样把 scheduler 断言从 < 改为 <= 并在 InputProcessor 增加超长偏移校验，其 prompt-end 回归测试被本 PR 吸收，作者 @meghaagr13 列为共同作者。
- PR #51891 Validate request indices and sampler sizes before engine work: 安全加固总纲（GHSA-98fv-r3w4-mrrv, Track B），其中 PTP-VLLM-111 'routed-expert prompt offset' 正是本 PR 落地的单项；本 PR 采用与其一致的 '提交前校验、拒绝非法值' 思路，但显式排除其余条目。