PR #41411 完整报告

vllm-project/vllm

[bugfix] Fix prompt logprobs on request eviction during chunked prefill

合并时间：2026-05-05 02:46

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/41411

执行摘要

- 一句话：修复 chunked prefill 请求驱逐时 prompt logprobs 丢失
- 推荐动作：值得精读这个小巧但关键的数据流修复。它展示了将临时状态与请求生命周期绑定的设计模式，并体现了如何通过边界调整和状态归属重构解决并发下的数据一致性问题。团队在类似 feature（如 speculative decoding 状态）的维护时可参考此模式。

功能与动机

PR body 明确指出：computed_prefill < prompt_lens - 1 检查错误地跳过了最后一个 prompt token，导致当该 token 恰好是 chunked prefill 的最后一个分块时，prompt logprobs 不会被计算。同时，将中间状态存储在 InputBatch 字典中在请求被驱逐后无法保留，必须绑定到请求对象自身。

实现拆解

1. 状态归属迁移（vllm/v1/worker/gpu_input_batch.py）：在 CachedRequestState 数据类中新增 in_progress_prompt_logprobs_cpu: LogprobsTensors | None 字段，用于跨 prefill 步骤累积 prompt logprobs 的 CPU 张量。移除 InputBatch 中的 in_progress_prompt_logprobs_cpu 字典，并在 remove_request 中删除对应清理代码，因为届时状态会随 CachedRequestState 自动回收。
2. 生产者适配（vllm/v1/worker/gpu_model_runner.py）：在 _get_prompt_logprobs_dict 方法中，将原先从 self.input_batch.in_progress_prompt_logprobs_cpu 获取字典的逻辑改为直接访问 request.in_progress_prompt_logprobs_cpu；条件判断从 logprobs_tensors = in_progress_dict.get(req_id) + if not logprobs_tensors 调整为 logprobs_tensors = request.in_progress_prompt_logprobs_cpu + if logprobs_tensors is None；完成预填充后通过 self.requests[req_id].in_progress_prompt_logprobs_cpu = None 清空而非 del in_progress_dict[req_id]。
3. 边界修复（vllm/v1/worker/gpu/sample/prompt_logprob.py）：在 compute_prompt_logprobs 中将 includes_prompt = computed_prefill < prompt_lens - 1 改为 includes_prompt = computed_prefill < prompt_lens，消除对最后一个 prompt token 的 one-off 错误。
4. 测试配套（tests/v1/e2e/general/test_async_scheduling.py, tests/conftest.py）：在 test_without_spec_decoding 和 test_with_eagle3_spec_decoding 的采样参数列表中加入 dict(prompt_logprobs=2) 与 dict(prompt_logprobs=2, logprobs=2)；修改

_logprobs_match 函数签名以支持 None，当任一参数为 None 时返回 相等性比较（lps_ais lps_b）；在 VllmRunner.generate 中将 prompt_logprobs 数据也扩展到 req_logprobs列表中。

关键文件：

- vllm/v1/worker/gpu_model_runner.py（模块 模型执行器；类别 source；类型 data-contract；符号 _get_prompt_logprobs_dict）：核心生产者：修改了 _get_prompt_logprobs_dict 方法，将状态访问从 InputBatch 字典切换到 CachedRequestState 字段，并调整了清理逻辑。
- vllm/v1/worker/gpu_input_batch.py（模块 输入批处理；类别 source；类型 core-logic；符号 CachedRequestState, InputBatch.init, InputBatch.remove_request）：状态定义和生命周期管理：在 CachedRequestState 中新增 in_progress_prompt_logprobs_cpu 字段，从 InputBatch 中移除对应字典，改变数据所有权。
- vllm/v1/worker/gpu/sample/prompt_logprob.py（模块 采样；类别 source；类型 core-logic；符号 PromptLogprobCompute.compute_prompt_logprobs）：直接修复 bug：修改 compute_prompt_logprobs 中的边界条件，确保最后一个 prompt token 也被包含。
- tests/v1/e2e/general/test_async_scheduling.py（模块 异步调度测试；类别 test；类型 test-coverage；符号 _logprobs_match, test_without_spec_decoding, test_with_eagle3_spec_decoding）：测试覆盖：增加了 prompt_logprobs=2 的配置到并发测试中，确保修复不被回归；修改 _logprobs_match 接受 None 值以正确对比。
- tests/conftest.py（模块 测试配置；类别 test；类型 test-coverage；符号 VllmRunner.generate）：测试基础设施适配：在 VllmRunner.generate 中将 prompt_logprobs 输出纳入 logprobs 对比列表，否则新增的 prompt_logprobs 测试无法正确验证。

关键符号：_get_prompt_logprobs_dict, PromptLogprobCompute.compute_prompt_logprobs, CachedRequestState.init, InputBatch.init, InputBatch.remove_request, _logprobs_match, VllmRunner.generate

关键源码片段

vllm/v1/worker/gpu_model_runner.py

核心生产者：修改了 _get_prompt_logprobs_dict 方法，将状态访问从 InputBatch 字典切换到 CachedRequestState 字段，并调整了清理逻辑。

```
def _get_prompt_logprobs_dict(
    self,
    hidden_states: torch.Tensor,
    num_scheduled_tokens: dict[str, int],
) -> dict[str, LogprobsTensors | None]:
    num_prompt_logprobs_dict = self.num_prompt_logprobs
    if not num_prompt_logprobs_dict:
        return {}
    # 不再从 InputBatch 中读取 in_progress_dict
    prompt_logprobs_dict: dict[str, LogprobsTensors | None] = {}
```

```
    completed_prefill_reqs = []
    for req_id, num_prompt_logprobs in num_prompt_logprobs_dict.items():
        num_tokens = num_scheduled_tokens.get(req_id)
        if num_tokens is None:
            continue # 请求被 preempted
        request = self.requests[req_id]
        if request.prompt_token_ids is None:
            continue # 不兼容 prompt embeddings
        num_prompt_tokens = len(request.prompt_token_ids)
        prompt_token_ids = torch.tensor(request.prompt_token_ids).to(
            self.device, non_blocking=True
        )
        # 直接从 request 取得或创建 logprobs_tensors
        logprobs_tensors = request.in_progress_prompt_logprobs_cpu
        if logprobs_tensors is None:
            logprobs_tensors = LogprobsTensors.empty_cpu(
                num_prompt_tokens - 1, num_prompt_logprobs + 1
            )
            request.in_progress_prompt_logprobs_cpu = logprobs_tensors
        # ... 后续计算逻辑不变 ...
    # 清理已完成的请求
    for req_id in completed_prefill_reqs:
        del num_prompt_logprobs_dict[req_id]
        self.requests[req_id].in_progress_prompt_logprobs_cpu = None # 原为 del in_progress_
        dict[req_id]
    return prompt_logprobs_dict
```

vllm/v1/worker/gpu/sample/prompt_logprob.py

直接修复 bug：修改 compute_prompt_logprobs 中的边界条件，确保最后一个 prompt token 也被包含。

```
class PromptLogprobCompute:
    # ...
    def compute_prompt_logprobs(
        self,
        # ... 参数列表
    ) -> dict[str, LogprobsTensors]:
        idx_mapping_np = input_batch.idx_mapping_np
        needs_prompt_logprobs = self.uses_prompt_logprobs[idx_mapping_np]
        if not np.any(needs_prompt_logprobs):
            return {}
        num_prompt_logprobs = self.num_prompt_logprobs[idx_mapping_np]
        prompt_lens = prompt_lens[idx_mapping_np]
        computed_prefill = num_computed_prefill_tokens[idx_mapping_np]
        # 关键修复：此前为 prompt_lens - 1，导致最后一个 token 的 logprob
        # 在 chunked prefill 的 final chunk 中被错误跳过
        includes_prompt = computed_prefill < prompt_lens
        resumed_after_prompt = prompt_lens < prefill_lens[idx_mapping_np]
        needs_prompt_logprobs &= includes_prompt & ~resumed_after_prompt
```

```
    if not np.any(needs_prompt_logprobs):
        return {}
    # ... 后续计算
```

评论区精华

与 reviewer njhill 的讨论：

- MRV2 一致性：njhill 询问 GPU model runner v2 是否也存在相同 bug。作者回复已在 MRV2 中验证无此问题。
- 测试增强：njhill 指出新增的测试可能会失败，因为需要修改 logprob 比较函数以处理 prompt logprobs 项。作者随后修改了 _logprobs_match 使其接受 None 并正确处理。
- 注释细节：njhill 建议新字段的注释保持简洁，仅保留 `# To accumulate prompt logprobs tensor chunks across prefill steps.` 即可，作者照做。
  - 检查 MRV2 是否受同一 bug 影响 (correctness): 确认 MRV2 无此 bug，无需额外修复。
  - 测试修改后失败，需要更新 logprobs 比较函数 (testing): 通过修改 _logprobs_match 和 generate 方法，测试通过。

风险与影响

- 风险：
  1. 状态迁移导致的内存泄漏：CachedRequestState 可能在请求结束后未被正确清理（如异常路径），若 in_progress_prompt_logprobs_cpu 持有大张量，可能造成内存泄漏。不过 PR 中在 completed_prefill_reqs 循环末尾显式将字段置为 None，降低了风险。
  2. 边界修订的回归：prompt_lens - 1 改为 prompt_lens 可能影响非 chunked prefill 或 spec decode 场景下的行为，但测试覆盖了这些组合，且 MRV2 中也同样修复（从讨论中得知），降低了回归概率。
  3. 测试覆盖局限：仅测试了 prompt_logprobs=2 和 prompt_logprobs=2, logprobs=2 两种配置，未覆盖 -1（全部 logprobs）或与 preemption 同时触发的边界情况，可能遗漏深层问题。 - 影响：直接影响在 V1 引擎中使用 prompt_logprobs 参数且发生 chunked prefill 或请求驱逐的用户：原先在这些场景下最后一个 prompt token 的 logprob 会缺失，修复后正确返回。不影响不使用 prompt_logprobs 的用户。对性能无显著影响，因为该功能本身很少被使用且代码路径保持简单。MRV2 用户不受此 PR 影响（因已在 MRV2 验证无此 bug）。 - 风险标记：状态所有权变更，边界修复风险，测试覆盖有限

关联脉络

- 暂无明显关联 PR