PR #45840 完整报告

vllm-project/vllm

[Perf] Skip/shrink all_token_ids copy in scheduler for non-async and V2 runner

合并时间：2026-06-21 06:36

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/45840

执行摘要

- 一句话：跳过非异步 /V2 调度器下 all_token_ids 复制
- 推荐动作：值得合并的低风险性能优化。实现简洁，测试充分，reviewer 已批准。对于关注调度性能或 V2 runner 迁移的团队，可精读调度器中的条件守卫变化。

功能与动机

CachedRequestData.all_token_ids 仅在 V1 GPU model runner 的异步调度分支中读取，但调度器之前每次调度步骤都会为所有恢复的请求复制完整 token 历史（prompt + output），并通过 msgpack 广播给所有 worker。对于 V2 runner 和非异步路径，这些数据从未使用，造成了不必要的序列化和内存复制开销。该 PR 旨在消除这部分冗余，同时保持 V1 异步路径的兼容性。

实现拆解

1. 在 vllm/v1/core/sched/scheduler.py 中，schedule() 方法内将 prev_step_scheduled_req_ids 的更新条件改为只在非 V2 runner（即 MRV1）时执行，避免 V2 下的无效记录。
2. 在 _make_cached_request_data() 中，将 all_token_ids 的填充逻辑嵌套在 if not self.use_v2_model_runner 条件下，仅当请求未在前一步被调度时才复制全量 req.all_token_ids；V2 runner 下该字典始终保持为空。
3. 在 vllm/v1/core/sched/output.py 中，更新 CachedRequestData.all_token_ids 的字段注释，明确标注为 MRV1-only。
4. 调整 V2 runner 分支中 scheduled_new_reqs 与 scheduled_resumed_reqs 的合并方式，由列表拼接变为 extend 加 clear，语义等价但更简洁。
5. 在 tests/v1/core/test_scheduler.py 中新增 test_cached_request_data_resumed_all_token_ids_mrv1_only，验证 V1 下传播全 token ids 而 V2 下为空；同时更新了两个现有测试，使其在默认非异步配置下预期 all_token_ids 为空。

关键文件：

- tests/v1/core/test_scheduler.py（模块 测试；类别 test；类型 test-coverage；符号 test_cached_request_data_resumed_all_token_ids_mrv1_only, make_cached）：新增针对性的单元测试，验证 V1 和 V2 下 all_token_ids 的正确行为，更新现有测试断言。

- vllm/v1/core/sched/scheduler.py（模块 调度器；类别 source；类型 core-logic）：核心调度器，修改了 _make_cached_request_data 和 schedule 方法，根据模型 runner 类型条件化 all_token_ids 的构建与传递。
- vllm/v1/core/sched/output.py（模块 数据结构；类别 source；类型 core-logic）：更新了 CachedRequestData.all_token_ids 的文档字符串，明确其为 MRV1-only。

关键符号：schedule, _make_cached_request_data, test_cached_request_data_resumed_all_token_ids_mrv1_only

关键源码片段

tests/v1/core/test_scheduler.py

新增针对性的单元测试，验证 V1 和 V2 下 all_token_ids 的正确行为，更新现有测试断言。

```
def test_cached_request_data_resumed_all_token_ids_mrv1_only():
    """all_token_ids carries a resumed request's token ids to the connector
    for the V1 model runner, but is skipped entirely for the V2 model runner.
    """
    from vllm.v1.core.kv_cache_manager import KVCacheBlocks

    scheduler = create_scheduler()
    (req,) = create_requests(num_requests=1, num_tokens=8)
    req.append_output_token_ids([101, 102, 103])

    # 确保请求在上一步未被调度（模拟恢复场景）
    assert req.request_id not in scheduler.prev_step_scheduled_req_ids

    empty_blocks = KVCacheBlocks(blocks=((),))

    def make_cached():
        return scheduler._make_cached_request_data(
            running_reqs=[],
            resumed_reqs=[req],
            num_scheduled_tokens={req.request_id: 1},
            spec_decode_tokens={},
            req_to_new_blocks={req.request_id: empty_blocks},
        )

    # V1 模型 runner：应保留全量 token ids
    assert not scheduler.use_v2_model_runner
    cached = make_cached()
    assert req.request_id in cached.resumed_req_ids
    assert cached.all_token_ids[req.request_id] == list(req.all_token_ids)

    # V2 模型 runner：all_token_ids 应被完全跳过
    scheduler.use_v2_model_runner = True
    cached = make_cached()
    assert req.request_id in cached.resumed_req_ids
```

```
    assert cached.all_token_ids == {}
```

vllm/v1/core/sched/scheduler.py

核心调度器，修改了 _make_cached_request_data 和 schedule 方法，根据模型 runner 类型条件化 all_token_ids 的构建与传递。

```
# 在 schedule() 方法末尾，记录本次被调度的请求 ID
# 仅 MRV1 需要此记录以分辨恢复请求
if not self.use_v2_model_runner:
    self.prev_step_scheduled_req_ids.clear()
    self.prev_step_scheduled_req_ids.update(num_scheduled_tokens.keys())

# 在 _make_cached_request_data 中构建 all_token_ids
if not self.use_v2_model_runner: # 字段仅 V1 模型 runner 会读取
    if req_id not in self.prev_step_scheduled_req_ids: # 恢复的请求
        # 复制完整的 token ids（包括 prompt 和 output），
        # 供 V1 异步调度分支恢复输出 token 列表
        all_token_ids[req_id] = req.all_token_ids.copy()
```

评论区精华

核心讨论围绕实现简化展开：njhill 评论“I don't think we need to optimize the MRV1 case; it's probably better to keep the full token ids in the resume case since I think this may be used by some OOT plugins”，建议仅对 V2 跳过，V1 保持原样以兼容外部插件。amanchugh89 接受建议，最终实现简化：V2 跳过，V1 保留全量。njhill 后续推入了简洁化提交并批准合并。

- 简化实现与 OOT 兼容性 (design): 采纳 njhill 建议：V1 异步路径全量复制，V2 及非异步路径跳过。
- 性能优化确认 (performance): njhill 批准合并。

风险与影响

- 风险：风险较低。V2 和非异步路径原本就不读取 all_token_ids，跳过其构建不会影响正确性；V1 异步路径行为与之前一致。主要风险在于外部插件可能依赖 all_token_ids 在 V1 非异步路径下的传递，但由于该字段在非异步路径过去也未被使用（gpu_model_runner.py 通过 use_async_scheduling 守卫），所以实际无影响。测试覆盖了关键场景。
- 影响：对使用 V2 model runner（默认）或非异步调度的用户，可减少调度步骤中约数千 token id 的复制和序列化开销（取决于请求历史长度），降低调度器与 worker 之间的通信负载。对 V1 异步调度用户无行为变化。代码库内调度模块和输出数据结构有轻微重构，但公共接口未变。
- 风险标记：低风险，V1 全 token ids 保留兼容 OOT，V2 序列化负载减少

关联脉络

- PR #34029 [Perf] Skip/shrink all_token_ids copy in scheduler for non-async and V2 runner: 前身 PR，已被关闭；此 PR 吸收了 review 反馈并最终实现。