PR #49356 完整报告

vllm-project/vllm

[CI][Bugfix] Fix and wire streaming-input tests

合并时间：2026-07-22 08:35

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/49356

执行摘要

- 一句话：修复并接入流式输入测试套件到 CI
- 推荐动作：建议合并。该 PR 修正了测试套件与代码的同步问题，并恢复了 CI 覆盖，是维护健康测试基础设施的必要步骤。值得关注的是调度器中 skipped_waiting 队列的使用，反映了流式输入会话的生命周期管理设计。

功能与动机

tests/v1/streaming_input/ 测试套件未运行于任何 CI job 中，已与代码脱节：mock 的 model_config 缺少 is_encoder_decoder/is_diffusion 属性导致调度器断言触发；调度器将暂停的流式会话移至 skipped_waiting 队列但断言仍检查 waiting；模型运行器缺少 late_interaction_runner 和 pp_handler 等新属性；且测试错误标记为 cpu_test 但实际需要 CUDA 设备。

实现拆解

1. 修复 test_scheduler_streaming.py：在 create_scheduler() 的 mock 中显式设置 is_encoder_decoder=False 和 is_diffusion=False，避免 MagicMock 的真值误判；将流式会话的队列断言从 scheduler.waiting 改为 scheduler.skipped_waiting。
2. 修复 test_gpu_model_runner_streaming.py：在 fixture 中增加 runner.late_interaction_runner = Mock()，满足 _update_streaming_request 的访问需求。
3. 修复 test_gpu_model_runner_v2_streaming.py：移除 RequestState 构造函数中已废弃的 model_dtype 和 cache_draft_logits 参数；增加 runner.pp_handler = None 以支持 _remove_request 的流式更新路径。
4. 移除错误的 cpu_test 标记：两个模型运行器测试因分配固定内存（UVA）需要 CUDA 设备，删除 pytestmark = pytest.mark.cpu_test 并添加注释说明原因。
5. 接入 CI：在 .buildkite/test_areas/misc.yaml 的 "V1 Core + KV + Metrics" job 的 source_file_dependencies 和 commands 中添加 tests/v1/streaming_input，使其在 GPU 上自动运行。

关键文件：

- tests/v1/streaming_input/test_scheduler_streaming.py（模块 调度器；类别 test；类型 test-coverage；符号 create_scheduler, test_streaming_e2e_lifecycle）：修复了 mock

缺失属性和断言队列错误，直接反映了调度器流式输入状态机的变更。

- tests/v1/streaming_input/test_gpu_model_runner_v2_streaming.py（模块 模型执行器；类别 test；类型 test-coverage；符号 mock_model_runner_with_req_states）：移除了已废弃的 RequestState 参数，并补充了 pp_handler 属性，反映了模型运行器的流式更新路径变更。
- tests/v1/streaming_input/test_gpu_model_runner_streaming.py（模块 模型执行器；类别 test；类型 test-coverage；符号 mock_model_runner_with_input_batch）：增加了 late_interaction_runner 属性并移除了错误的 cpu_test 标记。
- .buildkite/test_areas/misc.yaml（模块 CI 配置；类别 config；类型 configuration）：将 streaming_input 测试加入 CI 配置，确保这些测试在 GPU job 中自动运行。

关键符号：未识别

关键源码片段

tests/v1/streaming_input/test_scheduler_streaming.py

修复了 mock 缺失属性和断言队列错误，直接反映了调度器流式输入状态机的变更。

```
# tests/v1/streaming_input/test_scheduler_streaming.py

def create_scheduler() -> Scheduler:
    vllm_config = VllmConfig(device_config=DeviceConfig("cpu"))
    vllm_config.model_config = MagicMock()
    vllm_config.model_config.skip_tokenizer_init = True
    vllm_config.model_config.is_multimodal_model = False
    # 显式设置为 False，避免 MagicMock 的 truthy 值触发
    # 调度器中 is_encoder_decoder / is_diffusion 的断言
    vllm_config.model_config.is_encoder_decoder = False
    vllm_config.model_config.is_diffusion = False
    vllm_config.model_config.max_model_len = 1024
    vllm_config.model_config.enable_return_routed_experts = False
    # ... 其余代码不变
```

tests/v1/streaming_input/test_gpu_model_runner_v2_streaming.py

移除了已废弃的 RequestState 参数，并补充了 pp_handler 属性，反映了模型运行器的流式更新路径变更。

```
# tests/v1/streaming_input/test_gpu_model_runner_v2_streaming.py

# 不是 cpu_test: RequestState 会分配固定内存 (UVA),
# 即使 state 本身在 CPU 上，也需要 CUDA 设备

@pytest.fixture
def mock_model_runner_with_req_states():
    """Create a mock MRv2 GPUModelRunner with a real RequestState."""
    runner = Mock(spec=GPUModelRunner)
    runner.req_states = RequestState(
        max_num_reqs=10,
```

```
        max_model_len=1024,
        max_num_batched_tokens=1024,
        num_speculative_steps=0,
        vocab_size=32000,
        device=torch.device("cpu"),
        # model_dtype 和 cache_draft_logits 已从参数中移除
    )
    runner.encoder_cache = None
    runner.model_state = Mock()
    runner.block_tables = Mock()
    runner.lora_state = Mock()
    # _remove_request 会在流式更新路径中访问 pp_handler
    runner.pp_handler = None
    runner.sampler = None
    runner.prompt_logprobs_worker = None
    runner.is_last_pp_rank = False
    # ...
```

tests/v1/streaming_input/test_gpu_model_runner_streaming.py

增加了 late_interaction_runner 属性并移除了错误的 cpu_test 标记。

```
# tests/v1/streaming_input/test_gpu_model_runner_streaming.py

# 不是 cpu_test: InputBatch 会分配固定内存 (UVA),
# 即使 batch 张量在 CPU 上，也需要 CUDA 设备

@pytest.fixture
def mock_model_runner_with_input_batch():
    """Create a mock GPUModelRunner with a real InputBatch for e2e testing."""
    runner = Mock(spec=GPUModelRunner)
    runner.uses_mrope = False
    runner.requests = {}
    runner.max_num_reqs = 10
    runner.max_model_len = 1024
    # _update_streaming_request 会访问 late_interaction_runner
    runner.late_interaction_runner = Mock()
    # ...
```

评论区精华

无人工 review 讨论，仅由 Claude bot 评论指出 PR 来自 fork，自动审查被禁用。随后获得 sfeng33 的批准。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。变更仅限于测试代码和 CI 配置，无生产代码修改。但需注意：如果流式输入功能存在真实 bug，这些测试可能尚未覆盖所有场景，但至少保证了基础路径的回归保护。

- 影响：对用户无直接影响；对系统而言，流式输入测试将自动在 CI 中运行，防止未来回归；对团队而言，降低了维护成本并提高了测试覆盖率。
- 风险标记：测试覆盖修复

关联脉络

- 暂无明显关联 PR