

PR #53862 完整报告

vllm-project/vllm

[XPU][CI] increase timeout of extract_hidden_states tp2

合并时间: 2026-08-27 06:57

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/53862>

执行摘要

- 一句话: XPU 测试超时延至 240s, 修 CI 超时失败
- 推荐动作: 该 PR 体量极小, 不值得精读。可将其视为 XPU CI 稳定性维护的样例: 当发现平台相关测试超时且底层逻辑正确时, 按平台分支调整 `pytest.mark.timeout` 阈值即可。唯一值得注意的设计点是「显式平台分支 + 保持其它平台默认值」的写法, 既避免全局放宽超时掩盖真实性能回归, 又解决了特定硬件环境的慢启动问题。

功能与动机

PR body 明确说明目的是将 `extract_hidden_states` TP2 测试用例的超时从 120s 提升到 240s。结合 XPU CI 标签和 `current_platform.is_xpu()` 条件可以看出, XPU 环境上多卡初始化与模型加载耗时超出原先 120s 阈值, 导致集成测试偶发超时失败, 因此需要为 XPU 平台单独放宽限制。该改动没有关联 Issue, 属于典型的 CI 稳定性修复。

实现拆解

实现拆解如下:

1. 定位变更入口: 唯一改动文件为 `tests/v1/kv_connector/extract_hidden_states_integration/test_extraction.py`, 第 299-301 行的 `@pytest.mark.timeout` 装饰器。
2. 调整超时条件表达式: 将单行条件 `240 if current_platform.is_rocm() else 120` 改为三行展开的 `240 if current_platform.is_rocm() or current_platform.is_xpu() else 120`, 即 XPU 平台与 ROCm 一样获得 240s 超时窗口, 其余平台保持 120s。
3. 影响面: 该装饰器只作用于 `test_extract_hidden_states_tp2` 这一测试函数, 测试内容 (加载 Qwen3-0.6B、TP2 配置、hidden states 提取断言) 完全不变, 因此不影响任何源码路径或运行时行为, 仅改变 CI 执行阈值。
4. 测试配套与配置: 无需新增测试; 由于是测试本身的稳定性修正, 无独立验证步骤, 最终由 Buildkite CI (Build #85702) 和 reviewer 审核确认。

关键文件:

- `tests/v1/kv_connector/extract_hidden_states_integration/test_extraction.py` (模块 提取测试; 类别 test; 类型 test-coverage): 唯一变更文件, 将 `test_extract_hidden_states_tp2` 的超时条件从仅 ROCm 延长扩展为 ROCm/XPU 均延长到 240s, 是本次 PR 的全部内容。

关键符号：未识别

关键源码片段

[tests/v1/kv_connector/extract_hidden_states_integration/test_extraction.py](#)

唯一变更文件，将 `test_extract_hidden_states_tp2` 的超时条件从仅 ROCm 延长扩展为 ROCm/XPU 均延长到 240s，是本次 PR 的全部内容。

```
# tests/v1/kv_connector/extract_hidden_states_integration/test_extraction.py
# 关键变更：把 XPU 平台并入 240s 超时分支，避免 TP2 多卡初始化慢导致 CI 超时失败。
@pytest.mark.timeout(
    240 if current_platform.is_rocm() or current_platform.is_xpu() else 120
)
@multi_gpu_test(num_gpus=2)
@create_new_process_for_each_test()
def test_extract_hidden_states_tp2():
    """Test that hidden states extraction works with tensor_parallel_size=2."""
    tmp_dir = tempfile.mkdtemp()
    layer_ids = [5, 11, 17]
    hidden_size = 1024 # Qwen/Qwen3-0.6B hidden_size

    # 构造 TP2 的 LLM 实例：通过 speculative_config 开启 hidden states 提取，
    # 并通过 KV 连接器把提取结果写入共享存储，供后续断言校验。
    llm = LLM(
        model="Qwen/Qwen3-0.6B",
        tensor_parallel_size=2,
        speculative_config={
            "method": "extract_hidden_states",
            "num_speculative_tokens": 1,
            "draft_model_config": {
                "hf_config": {"eagle_aux_hidden_state_layer_ids": layer_ids}
            },
        },
        kv_transfer_config={
            "kv_connector": "ExampleHiddenStatesConnector",
            "kv_role": "kv_producer",
            "kv_connector_extra_config": {"shared_storage_path": tmp_dir},
        },
        max_model_len=256,
        enforce_eager=True,
        gpu_memory_utilization=0.4,
        load_format="dummy",
    )

    prompts = ["Hello world", "Test prompt with several tokens"]
    sampling_params = SamplingParams(max_tokens=1, temperature=0.0)
    outputs = llm.generate(prompts, sampling_params)

    # 校验每个输出都携带 hidden_states_path，且从共享存储读回的张量形状正确。
```

```
assert len(outputs) == len(prompts)
for output in outputs:
    assert output.kv_transfer_params is not None
    hidden_states_path = output.kv_transfer_params.get("hidden_states_path")
    assert hidden_states_path is not None

    obj = example_hidden_states_connector.load_hidden_states(hidden_states_path)
    token_ids = obj["token_ids"]
    hidden_states = obj["hidden_states"]

    assert torch.equal(token_ids, torch.tensor(output.prompt_token_ids))
    assert hidden_states.shape == (
        len(output.prompt_token_ids),
        len(layer_ids),
        hidden_size,
    )
```

评论区精华

该 PR 的 review 交互非常简洁：作者 [zhenwei-intel](#) 请求 [@jikunshang](#) 审核，[jikunshang](#) 触发 `/ci run` 并通过 Buildkite CI 验证后直接批准（APPROVED）。[claude\[bot\]](#) 因该 PR 来自 fork 而自动跳过 review，没有产生实质代码讨论。没有出现关于超时值选取、是否应统一为 240s、或底层 XPU 性能问题的争议，所有疑虑已在 CI 跑通和 approve 后闭环。

- 暂无高价值评论线程

风险与影响

- 风险：技术风险极低：
 1. 回归风险：仅修改 pytest 装饰器中的超时条件，不触碰测试断言、模型配置或连接器逻辑，不会影响功能正确性。
 2. CI 盲区风险：将 XPU 超时放宽到 240s 后，若后续 XPU 性能进一步劣化，测试失败会从「超时」变为「更晚的超时」，但并不会掩盖真正失败；真正的功能失败仍会被断言捕获。
 3. 平台覆盖风险：条件写法为 `is_rocm()` or `is_xpu()`，与其它平台保持 120s；若未来新增同样慢的平台，需要再次在此处同步更新，存在小幅维护隐患，但不影响当前正确性。
 4. 不确定性说明：由于没有展示 XPU 具体超时统计数据，无法确认 240s 是否长期充分，只能视为基于 CI 观察的工程性修补。- 影响：影响范围非常有限：
- 用户影响：无。该 PR 不改变任何模型推理、前端 API 或 KV 连接器行为。
- 系统影响：仅影响 XPU CI 任务中 `extract_hidden_states_tp2` 的测试执行时长上限，避免 XPU 机器因初始化慢而被误判为失败。
- 团队影响：降低 XPU 维护者在 kv-connector 集成测试上的 CI 噪音，缩短排查非确定性超时的时间成本。属于低影响、低收益的稳定性维护。
- 风险标记：仅调整测试超时，平台条件需持续维护，缺少超时统计依据

关联脉络

- PR #55055 [BugFix][Model Runner V2][Spec Decode] Fix decode instance's multi-layer MTP kv caches during P/D: 同一 tests/v1/kv_connector/ 测试目录下与 speculative decoding 的 hidden states 提取相关的 kv-connector 测试, 本 PR 正是为该集成测试线调整 XPU 超时。
- PR #57041 [Config] Infer HiSparse attention config from HiSparseConnector: 同为 kv-connector 模块的配置与测试改动, 反映 kv-connector 集成测试是近期活跃维护区域。