PR #52608 完整报告

vllm-project/vllm

[Bugfix][CI] Release the shared ColBERT engine before `test_colbert_hf_comparison`

合并时间：2026-08-18 01:50

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/52608

执行摘要

- 一句话：class 作用域提前释放共享 engine，修复 ColBERT 测试 VRAM 回归
- 推荐动作：值得精读，尤其适合负责 GPU CI 稳定性的同学。核心学习点是 pytest fixture scope 的 teardown 时机对显存敏感测试的影响：module 作用域 fixture 的释放太晚，会与后续自建 runner 的测试产生 VRAM 竞争；class 作用域则把释放提前到类结束。另一个可借鉴的设计是 " 共享 engine 用例收进类、自建 runner 用例留在模块级 " 的混合编排，既有提速又不牺牲隔离性。建议后续观察是否需要一个更显式的 engine 释放机制，而不是依赖 scope 顺序这一隐式约定。

功能与动机

PR body 明确指出失败由 #52570 引入，错误信息为 ValueError: Memory of devices devices=[0] not free after dur_s=240.14。根因是模块级 fixture 只在模块结束时 finalize，最后一个参数化的 engine 仍占住 gpu_memory_utilization（默认 0.92）的设备内存，而新启动 engine 需要同样的空闲比例才能创建，因此第二个 runner 永远无法启动；且该条件是比例性的，设备再大也无济于事。这些测试又不能直接共享 fixture 的 engine，因为 HF 对比需要 float32 而 fixture 使用 half，且每个用例还会固定设置 VLLM_USE_V2_MODEL_RUNNER。

实现拆解

整个变更集中在 tests/models/language/pooling/test_colbert.py 一个文件，共 3 个 commit，其中两个是合并 main 分支，实质改动只有第一个 commit。

1. 调整 fixture 作用域：colbert_spec、colbert_model_name、colbert_dim、colbert_max_model_len、colbert_extra_kwargs、colbert_model 六个 fixture 全部从 scope="module" 改为 scope="class"。
2. 重组测试结构：原本模块级的六个共享 engine 的测试函数 test_colbert_token_embed、test_colbert_late_interaction_1_to_1、test_colbert_late_interaction_1_to_N、test_colbert_late_interaction_N_to_N、test_colbert_relevance_ordering、test_colbert_embed_not_supported 被收进新增的 TestColbertSharedEngine 类中，转为类方法。
3. 保留独立用例：test_colbert_hf_comparison 继续作为模块级参数化测试存在，因为它需要自建 runner（float32 + 各自固定 VLLM_USE_V2_MODEL_RUNNER），且 pytest 保证 class-scoped fixture 在类结束时先被 finalize，之后才运行模块级测试，从而让 engine

在比较测试启动前必然释放。

4. 配套验证：PR 给出修复前后整文件与单独运行的对比数据，并说明四个失败只在整文件运行时确定性复现，从行为上印证了共享 engine 是根因。该 PR 还进入了 v0.28.0 cherry picks 里程碑，后续需同步到发布分支。

关键文件：

- tests/models/language/pooling/test_colbert.py（模块 测试编排；类别 test；类型 test-coverage；符号 test_colbert_token_embed, test_colbert_late_interaction_1_to_1, TestColbertSharedEngine, test_colbert_late_interaction_1_to_N）：唯一变更文件。将六个共享 engine 的测试从模块级函数重构为 TestColbertSharedEngine 类方法，并把相关 fixtures 从 scope="module" 改为 scope="class"，利用 pytest 在类结束时先 finalize class-scoped fixture 的语义，确保 engine 在 test_colbert_hf_comparison 自建 runner 启动前释放。

关键符号：colbert_model, colbert_spec, TestColbertSharedEngine, test_colbert_hf_comparison

关键源码片段

tests/models/language/pooling/test_colbert.py

唯一变更文件。将六个共享 engine 的测试从模块级函数重构为 TestColbertSharedEngine 类方法，并把相关 fixtures 从 scope="module" 改为 scope="class"，利用 pytest 在类结束时先 finalize class-scoped fixture 的语义，确保 engine 在 test_colbert_hf_comparison 自建 runner 启动前释放。

```
# 六个共享 engine 的测试统一收进类作用域，
# 配套 fixtures 由 module 作用域改为 class 作用域。
# pytest 会在类结束时先 finalize class-scoped fixtures，
# 再继续运行模块级测试，从而保证 engine 释放早于
# test_colbert_hf_comparison 自建 runner 的启动。
@pytest.fixture(params=list(COLBERT_MODELS.keys()), scope="class")
def colbert_spec(request):
    """返回当前参数化对应的模型规格 dict。"""
    return COLBERT_MODELS[request.param]


# 每个 model 只启动一次 engine，保留 #52570 的启动优化。
@pytest.fixture(scope="class")
def colbert_model(
    vllm_runner,
    colbert_model_name,
    colbert_max_model_len,
    colbert_extra_kwargs,
):
    with vllm_runner(
        colbert_model_name,
        runner="pooling",
```

```
        dtype=DTYPE,
        max_model_len=colbert_max_model_len,
        enforce_eager=True,
        **colbert_extra_kwargs,
    ) as vllm_model:
        yield vllm_model


class TestColbertSharedEngine:
    """共享 engine 的测试。

    类作用域保证 engine 在 test_colbert_hf_comparison 之前释放，
    否则该测试自建 runner 会因 VRAM 被占用而无法启动。
    """

    def test_colbert_token_embed(self, colbert_model, colbert_dim):
        """验证 ColBERT 模型产出 token embeddings。"""
        outputs = colbert_model.token_embed([TEXTS_1[0]])
        assert len(outputs) == 1
        emb = torch.as_tensor(outputs[0])
        assert emb.dim() == 2
        assert emb.shape[1] == colbert_dim
        assert emb.shape[0] > 1

    def test_colbert_embed_not_supported(self, colbert_model):
        """ColBERT 模型不支持 embed 任务，预期抛出 ValueError。"""
        with pytest.raises(ValueError, match="Embedding API is not supported"):
            colbert_model.embed([TEXTS_1[0]])
```

评论区精华

本 PR 的 review 环节没有实质性的技术交锋：claude[bot] 提示该 PR 来自 fork、自动化 review 被禁用，需要维护者手动处理；AndreasKaratzas 与 mgoin 先后 APPROVED，AndreasKaratzas 在合并前两次通过 /ci run 触发 Buildkite CI（构建 #84209 与 #84223）验证。技术上的关键设计权衡写在 PR body 中：为什么不能直接共享 fixture 的 engine —— HF 对比要求 rtol=atol=1e-2 精度、必须用 float32，而共享 fixture 是 half；且四个比较用例各自固定 VLLM_USE_V2_MODEL_RUNNER 取值，无法复用同一 runner。因此采用 " 类作用域共享 + 模块级独立 runner" 的组合，而不是退回 #52570 之前的每测试一个 runner 的慢路径。

- fork PR 的自动化 review 不可用 (other): 维护者 AndreasKaratzas 与 mgoin 手动 review 后均 APPROVED，并完成合并。
- CI 验证与调度 (other): CI 验证通过后 PR 被合并，并纳入 v0.28.0 cherry picks 里程碑。

风险与影响

- 风险：风险总体很低，因为是纯测试文件变更，不触及任何生产代码路径。需要留意的点包括：

1. 测试结构耦合：TestColbertSharedEngine 类内的所有测试现在隐式依赖共享 engine，未来若有人往该类里新增不需要 engine 的轻量测试，也会被迫触发 engine 启动，浪费资源。
2. fixture 作用域顺序依赖：修复的成立依赖 pytest "class-scoped fixture 在类结束时先 finalize、再执行模块级测试 " 的语义；如果未来有人在 test_colbert_hf_comparison 之前又插入其他模块级且需要自建 runner 的测试，同样的 VRAM 竞争可能再次出现，这只是一个局部修复而非通用的资源释放机制。
3. 参数化放大：colbert_spec 按 COLBERT_MODELS 所有 key 参数化，每个参数化值都会让类内全部测试重跑一次；若后续模型列表变大，整文件耗时仍会线性增长。 - 影响：对用户零影响，不涉及推理、调度或任何产品行为。对 CI 系统影响显著：消除了 4 个确定性失败的 test_colbert_hf_comparison 用例，整文件执行时间从 17m22s 缩短到 2m57s，去掉了四段以超时告终的 VRAM 等待，为每次 CI 运行节省约 14 分钟。对团队而言，该修复被纳入 v0.28.0 cherry picks 里程碑，意味着需要同步回发布分支；同时它明确了 vLLM GPU 测试中 fixture 作用域与显存生命周期的编排约定。 - 风险标记：测试结构耦合，依赖 fixture 作用域顺序，无生产代码变更

关联脉络

- PR #52570 [CI] Speed up ColBERT pooling tests by sharing engine per model（标题依据本 PR body 推断）：本 PR 直接修复 #52570 引入的回归：#52570 用模块级 colbert_model fixture 替换了 per-test runner，节省启动开销，但模块结束时才 finalize engine，阻塞了后续 test_colbert_hf_comparison 的自建 runner。
- PR #52763 [ROCM][CI] Attention test speedup: 同属 CI 测试提速与稳定性主题，说明团队在系统性缩短测试时长，本 PR 是其中一环的回归修复。
- PR #52810 [CI][ROCm] Prevent Git maintenance races during shallow fetches: 同属 CI 稳定性修复方向，均是为消除确定性竞态而做的窄范围修复。