PR #50284 完整报告

vllm-project/vllm

[CI] Stabilize speculator memory teardown

合并时间：2026-07-30 18:42

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/50284

执行摘要

- 一句话：使用 managed runner 稳定 spec 测试内存清理
- 推荐动作：值得快速合入以稳定 ROCm CI。代码风格简洁，无架构影响。

功能与动机

ROCm CI 中 speculator 测试因 GPU 内存未在下一个引擎启动前完成 teardown 而失败，参考 PR body 中的 Buildkite 构建链接。

实现拆解

1. 引入 vllm_runner fixture：在 test_speculators_model_integration 函数签名中添加 vllm_runner 参数，该 fixture 是 vllm 测试基础设施提供的 managed runner。
2. 替换 spec_llm 构造：将 LLM(...) 直接构造改为 with vllm_runner(...) as spec_runner: 上下文管理器，确保退出作用域时自动清理引擎、释放 GPU 内存。
3. 替换 ref_llm 构造：对 reference engine 采用同样的上下文管理器模式，自动执行 teardown。
4. 移除手动清理代码：删除 del spec_llm、del ref_llm、torch.accelerator.empty_cache() 和 cleanup_dist_env_and_memory() 调用，这些现在由 vllm_runner 的 __exit__ 处理。

关键文件：

- tests/v1/e2e/spec_decode/test_spec_decode.py（模块 推测解码；类别 test；类型 test-coverage；符号 test_speculators_model_integration）：唯一修改的文件，将 speculator 模型集成测试中的 LLM 构造改为使用 managed runner，确保 ROCm 上 GPU 内存正确释放。

关键符号：test_speculators_model_integration

关键源码片段

tests/v1/e2e/spec_decode/test_spec_decode.py

唯一修改的文件，将 speculator 模型集成测试中的 LLM 构造改为使用 managed runner，确保 ROCm 上 GPU 内存正确释放。

```
# tests/v1/e2e/spec_decode/test_spec_decode.py
# 关键变更：使用 vllm_runner 上下文管理器自动管理 LLM 生命周期
```

```
# 之前：手动构造 LLM，手动 del 和清理缓存
# 之后：with vllm_runner(...) as runner: 退出时自动清理

def test_speculators_model_integration(
    monkeypatch: pytest.MonkeyPatch,
    sampling_config: SamplingParams,
    model_path: str,
    expected_accuracy_threshold: float,
    vllm_runner, # 新增 fixture 参数
):
    monkeypatch.setenv("VLLM_ALLOW_INSECURE_SERIALIZATION", "1")
    test_prompts = get_test_prompts(mm_enabled=False)

    # First run: Direct speculator model (simplified integration)
    with vllm_runner(
        model_path, trust_remote_code=False, enable_chunked_prefill=None,
        compilation_config=CompilationConfig(), max_model_len=4096,
        gpu_memory_utilization=0.92,
    ) as spec_runner:
        evaluate_llm_for_gsm8k(spec_runner.llm, expected_accuracy_threshold)
        spec_outputs = spec_runner.llm.chat(test_prompts, sampling_config)
        # ... 断言保持不变 ...
        verifier_model = spec_runner.llm.llm_engine.vllm_config.model_config.model

    # Second run: Reference without speculative decoding
    with vllm_runner(
        verifier_model, trust_remote_code=False, enable_chunked_prefill=None,
        compilation_config=CompilationConfig(), max_model_len=4096,
        gpu_memory_utilization=0.92,
    ) as ref_runner:
        ref_outputs = ref_runner.llm.chat(test_prompts, sampling_config)

    # 比较输出 - 逻辑不变
    matches = sum(...)
    assert matches >= int(0.66 * len(ref_outputs))
```

评论区精华

无 review 评论，仅 claude[bot] 自动检测到 fork 模式并提示，tjtanaa 直接批准。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低。变更仅涉及测试文件，将手动管理 LLM 生命周期改为使用经过良好测试的 vllm_runner fixture，逻辑等效。如果 vllm_runner 在某种 corner case 下 cleanup 不完整，可能掩盖真实问题，但概率很小。
- 影响：仅影响 test_speculators_model_integration 一个测试函数，预期在 ROCm CI 上减少 flaky 失败。对其他平台无影响。

- 风险标记：仅测试变更

关联脉络

- PR #50000 [New model] Kimi K3: 共享 spec_decode 测试文件，可能涉及类似的内存清理模式
- PR #50340 [CI][ROCm] Stabilize LLM GC teardown check: 同为 ROCm CI 稳定性修复，方向一致