

PR #49910 完整报告

vllm-project/vllm

[CI] Explicitly tear down speculative decode runners

合并时间: 2026-07-27 13:32

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/49910>

执行摘要

- 一句话: CI 测试中显式关闭推测解码 runner
- 推荐动作: 值得合入 — 改进 CI 健壮性, 无副作用。

功能与动机

在 AMD CI 运行中, 参数化的 max-length 测试由于引擎未显式关闭, 导致后续测试可能受前一个测试的引擎状态影响 (如 GPU 内存不足或配置污染)。PR body 引用了 Buildkite 日志链接, 表明需要显式关闭引擎以隔离测试环境。

实现拆解

1. 修改 `test_ngram_max_len`: 函数签名新增 `vllm_runner` 参数, 将 `LLM(...)` 调用替换为 `with vllm_runner(...) as runner:`, 所有生成操作通过 `runner.llm.generate()` 完成。
2. 修改 `test_eagle_max_len`: 同样新增 `vllm_runner` 参数, 将 `LLM(...)` 调用嵌入 `vllm_runner` 上下文管理器, 生成逻辑相应调整。
3. 清理导入: 移除顶部未使用的 `from vllm import LLM` 导入, 依赖仅由 `vllm_runner` 提供。

关键文件:

- `tests/v1/spec_decode/test_max_len.py` (模块 测试; 类别 `test`; 类型 `test-coverage`; 符号 `test_ngram_max_len`, `test_eagle_max_len`): 唯一改动文件; 使用 `vllm_runner` fixture 替换直接 `LLM` 实例化, 确保引擎在参数化测试间被正确关闭。

关键符号: `test_ngram_max_len`, `test_eagle_max_len`

关键源码片段

`tests/v1/spec_decode/test_max_len.py`

唯一改动文件; 使用 `vllm_runner` fixture 替换直接 `LLM` 实例化, 确保引擎在参数化测试间被正确关闭。

```
# 变更后: test_ngram_max_len 使用 vllm_runner 上下文管理器 # 确保引擎在每个参数化实例后释放, 防止资源泄露
@pytest.mark.parametrize("num_speculative_tokens", [1, 3, 10])
def test_ngram_max_len(num_speculative_tokens: int, vllm_runner):
    with vllm_runner(
        "facebook/opt-125m",
        trust_remote_code=False,
        max_model_len=100,
        enable_chunked_prefill=None,
        enforce_eager=True, #
```

```
For faster initialization. speculative_config={ "method": "ngram", "
prompt_lookup_max": 5, "prompt_lookup_min": 3, "
num_speculative_tokens": num_speculative_tokens, }, ) as runner:
sampling_params = SamplingParams(max_tokens=100, ignore_eos=True)
runner.llm.generate(_PROMPTS, sampling_params) # 变更后: test_eagle_max_len 同样使
用 vllm_runner 上下文管理器 @pytest.mark.parametrize("num_speculative_tokens", [1,
3, 10]) @pytest.mark.parametrize("attn_backend",
get_attn_backend_list_based_on_platform()) def test_eagle_max_len( monkeypatch:
pytest.MonkeyPatch, num_speculative_tokens: int, attn_backend: str,
vllm_runner, ): if attn_backend == "ROCM_AITER_FA" and
current_platform.is_rocm(): monkeypatch.setenv("VLLM_ROCM_USE_AITER", "1")
with vllm_runner( "meta-llama/Meta-Llama-3-8B-Instruct",
trust_remote_code=False, enable_chunked_prefill=None,
enforce_eager=True, speculative_config={ "method": "eagle", "
model": "yuhuali/EAGLE-LLaMA3-Instruct-8B", "num_speculative_tokens":
num_speculative_tokens, "max_model_len": 80, },
max_model_len=200, attention_config={"backend": attn_backend}, ) as runner:
# ... 测试逻辑不变
```

评论区精华

无 review 讨论。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低：纯测试变更，不影响生产代码；仅修改 fixture 使用方式，保持测试语义不变。
- 影响：影响局限在 CI 测试阶段，特别是 AMD CI 上的 max-length 推测解码测试，提高了测试隔离性和可靠性。对其他硬件平台和测试无影响。
- 风险标记：暂无

关联脉络

- 暂无明显关联 PR