

PR #54934 完整报告

vllm-project/vllm

[CI][CPU] Add speculative-decoding coverage to CPU CI

合并时间: 2026-09-14 19:51

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/54934>

执行摘要

- 一句话: 为 CPU 平台新增投机解码 CI 步骤与 E2E/Smoke 测试
- 推荐动作: 值得中等优先级精读, 尤其是负责 CI 与 CPU 后端的工程师。关注三个设计决策: 一是 ngram e2e 用 drafted/accepted 指标做“反空转”断言, 避免用例在 proposer 不出草稿时仍然通过; 二是 smoke 用例断言 use_v2_model_runner 为真并要求接受长度下限, 把“功能回退”和“性能回退”都纳入门禁; 三是 soft_fail + source_file_dependencies 的上线策略。建议评审者跟进: 把 MIN_ACCEPTANCE_LEN 与 checkpoint 版本一起记录、明确软失败转硬门禁的条件、确认 VLLM_CPU_KVCACHE_SPACE 在两处设置下的真实取值。

功能与动机

PR body 指出: CPU 投机解码由 #32662 落地, 引入了 csrc/cpu/spec_decode_utils.cpp 与 vllm/utils/cpu_triton_utils.py, 后者被 CPUModelRunner._postprocess_triton 在 Triton-CPU 不可用时替换 Triton 内核; 但“nothing in tests/ or .buildkite/ referenced them, on any platform”。维护者 bigPYJ1151 在讨论中把诉求从内核单测改到端到端门禁: “adding some E2E tests is more helpful for gating SD functionality and correctness on CPU”, 并明确 GSM8K 精度类测试对现有 CPU CI 太贵, “we can only enable them as nightly tests in the future”, 因此本 PR 的目标是低成本地建立 CPU SD 的持续验证与功能矩阵声明。

实现拆解

1. 测试入口: 新增 tests/v1/e2e/test_cpu_spec_decode.py (+167)。文件级用 `pytestmark = pytest.mark.cpu_model` 加 `current_platform.is_cpu()` 收集期跳过, 保证跨套件收集时不会误跑; `os.environ.setdefault("VLLM_CPU_KVCACHE_SPACE", "1")` 只在本地生效 (CI 侧脚本已注入 16)。核心辅助符号 `_make_llm` (Qwen/Qwen3-0.6B、bfloat16、`disable_log_stats=False` 以便读指标)、`_tuples` (把输出归一成 token_ids/text/logprobs 三元组)、`baseline_refs` (module 级 fixture, 只跑一次无投机基线)、`_spec_metric` (从 `llm.get_metrics()` 按名字取 `vllm:spec_decode_num_draft_tokens / num_accepted_tokens`, 取不到返回 0.0)。
2. 两个门禁用例。`test_ngram_spec_decode_matches_baseline` 先用 `drafted > 0` 与 `accepted > 0` 做反“空转”断言, 再用 `check_logprobs_close` 比对基线与 ngram 投机输出, 保证 greedy 结果不被投机改变; `test_v2_speculator_smoke` 对 eagle3 / dflash / mtp 三组 `SMOKE_CONFIGS` 参数化, 先断言 `use_v2_model_runner` 为真以防静默回退 V1,

再用 `compute_acceptance_len` 断言接受长度不低于 `MIN_ACCEPTANCE_LEN = 1.5` (CPU 实测 1.85 / 1.99 / 2.29) , 并在 `finally` 中释放 LLM。

3. CI 步骤注册: `.buildkite/hardware_tests/cpu.yaml` (+46) 。新增 CPU-Spec Decode Tests (key `cpu-spec-decode-tests`、`device: intel_cpu`、`no_plugin: true`、`soft_fail: true`) , 用 `source_file_dependencies` 限定在 `vllm/v1/spec_decode/`、`vllm/v1/worker/cpu/`、`vllm/config/speculative.py`、`qwen3_5 / llama_eagle3` 等模型与 `mamba` 相关路径及测试目录, `commands` 分两段: 先跑 14 个单测, 再 `unset VLLM_CPU_CI_ENV` 后单独跑 `e2e` ; `soft_fail` 为观察期策略, 不阻塞合入。
4. CI 脚本与编译路径配套。 `.buildkite/scripts/hardware_ci/run-cpu-test.sh` 把 `OFFLINE_RETRY_PATTERN` 从只匹配 `huggingface_hub` 原始异常类, 扩到同时匹配 `Invalid repository ID or local directory specified`, 因为 `vllm` 的 `get_config()` (`transformers_utils/config.py`) 会把离线异常包装成通用 `ValueError`, 否则配置查询类缓存缺失永远触发不了在线回退; `commit d9ba3b6` 又去掉 `_make_llm` 里的 `enforce_eager=True`, 并在 `e2e` 前 `unset` CI 专用环境变量, 让用例真正走 CPU 用户生产默认的 `CompilationMode.VLLM_COMPILE (inductor)` 路径。
5. 文档矩阵。 `docs/features/README.md` 把 SD 行的 CPU 单元格由 ❷ 改为 ❸, 与新增 CI 覆盖一致; 这一步是声明而非运行时改动, 因此必须与 `soft_fail` 的现状一起理解。

关键文件:

- `tests/v1/e2e/test_cpu_spec_decode.py` (模块 推测解码; 类别 `test`; 类型 `test-coverage` ; 符号 `_make_llm`, `_tuples`, `baseline_refs`, `_spec_metric`) : 新增的 CPU 投机解码 E2E/smoke 用例文件, 是本 PR 的核心资产: 定义了度量取法、反空转断言、V2 runner 校验与接受长度下限。
- `.buildkite/hardware_tests/cpu.yaml` (模块 CI 流水线; 类别 `infra`; 类型 `configuration`; 符号 CPU-Spec Decode Tests, `cpu-spec-decode-tests`) : 注册 CPU-Spec Decode Tests 步骤, 决定触发条件、超时、软失败策略与两段式命令 (先单测、后 `e2e`) , 是让测试真正进入 CI 的关键配置。
- `.buildkite/scripts/hardware_ci/run-cpu-test.sh` (模块 CI 脚本; 类别 `infra`; 类型 `infrastructure`; 符号 `OFFLINE_RETRY_PATTERN`) : 放宽 HF 离线回退匹配, 使 `config` 查询类缓存缺失能触发一次在线重试, 是新 `e2e` 需要的模型下载前置条件。
- `docs/features/README.md` (模块 功能矩阵; 类别 `docs`; 类型 `documentation`) : 功能矩阵中 SD 行的 CPU 单元格由 ❷ 改为 ❸, 对外声明 CPU 支持投机解码, 需与 CI 仍处 `soft_fail` 的现状对齐理解。

关键符号: `test_ngram_spec_decode_matches_baseline`, `test_v2_speculator_smoke`, `_make_llm`, `_tuples`, `_spec_metric`, `baseline_refs`, `OFFLINE_RETRY_PATTERN`, CPU-Spec Decode Tests

评论区精华

讨论主要集中在“测什么”和“怎么控制成本”两条线。首轮 bigPYJ1151 直接否掉了原方案: “considering vLLM CPU already start to use model runner V2 and Triton kernels for operations in the model runner, I think it's not worth to add tests for the original

kernels”，作者接受并改为实测 V2 speculator 单测在 CPU-only 机器上的通过情况。第二轮维护者把要求升级为 engine 级门禁，并指出既有 tests/v1/e2e/spec_decode/ “are relatively CUDA specific”，而 GSM8K 精度测试“too expensive for current CPU CI capability”。围绕 V1/V2 覆盖边界，作者明确“this E2E gates V1, not V2”，因为 ngram 在 V2 unsupported 列表、suffix 不在 allowlist，其余方法都要 draft checkpoint。MTP 用例的成本假设也被推翻：commit fa59e0a 记录“The previously assumed ~55min CPU Triton-autotune cost for a hybrid MTP target didn't reproduce”，Qwen3.5 的 GDN 层走 CPU 原生算子、无 Mamba2-SSD 层可 autotune，于是加入 Qwen/Qwen3.5-0.8B-Base 并同时把步骤设为 soft_fail。

- 是否为 CPU 原有 spec decode kernel 补测试 (design): 放弃针对 csrc/cpu/spec_decode_utils.cpp 的内核用例，改为注册 V2 speculator 单测 + engine 级 e2e；最终 CI 步骤中的 14 个单测文件与当初实测跑通的集合一致。
- 新 e2e 不复用 tests/v1/e2e/spec_decode/，以及 V1/V2 覆盖边界 (testing): 采用自建小 e2e (ngram 覆盖 V1 engine 路径) + V2 speculator 单测 /smoke 的组合覆盖；V2 侧 e2e 由后续 MTP smoke 补齐。
- MTP smoke 用例选型与 hybrid Triton autotune 成本 (performance): SMOKE_CONFIGS 加入 mtp 参数项（接受长度 2.29，用例一分钟内跑完），并同步把整个步骤设为 soft_fail。
- CI 步骤先软失败上线并观察抖动 (other): 步骤上线即不阻塞合入，由维护者多日观察；但由软失败转为硬门禁的时间点与判定标准未在 PR 中约定。
- e2e 打开 torch.compile 覆盖生产实际路径 (testing): e2e 在真实 inductor 后端下运行，覆盖 CPU 用户默认的编译模式；同时意味着该步骤对编译相关回归更敏感，也更容易受编译耗时影响。

风险与影响

- 风险：1) 软失败掩盖回归：.buildkite/hardware_tests/cpu.yaml 中 soft_fail: true，步骤失败不会阻塞合入，维护者计划“monitor them for days”，但观察期退出条件未在 PR 中写明，存在长期停留在软失败、覆盖率虚高的风险。2) 阈值与模型硬编码抖动：MIN_ACCEPTANCE_LEN = 1.5 相对 eagle3 实测 1.85 只留 0.35 余量，SMOKE_CONFIGS 又绑定 meta-llama / nm-testing / z-lab / Qwen 四方 checkpoint，上游改权重或删除仓库会直接让 CPU CI 变红。3) CI 脚本回退匹配过宽：OFFLINE_RETRY_PATTERN 新增的 Invalid repository ID or local directory specified 是子串匹配，可能把真实配置 / 路径错误误判为可重试的缓存缺失，多触发一次在线尝试并延长失败用例耗时。4) 环境变量语义不一致：测试内 os.environ.setdefault("VLLM_CPU_KV_CACHE_SPACE", "1") 的注释意图是限制内存，但 run-cpu-test.sh 已注入 VLLM_CPU_KV_CACHE_SPACE=16，CI 中该 setdefault 不生效，内存占用实际由脚本决定。5) 内部接口耦合：smoke 用例直接断言 llm.llm_engine.vllm_config.use_v2_model_runner，一旦 LLM/Engine 暴露方式调整，测试会以“非功能性”原因失败。
- 影响：对最终用户无运行时行为变化，唯一直接影响是功能矩阵把 CPU 的 SD 标记为支持，可能抬高用户预期，而实际 CI 仍是 soft_fail 状态。对系统与 CI：intel_cpu 硬件队列新增一个 30 分钟超时预算的步骤（单测 + 4 个用例的 e2e），会占用 CPU CI 资源与 HF 缓存 / 带宽；同时 e2e 步骤 unset VLLM_CPU_CI_ENV 后走真实 inductor 路径，使 CPU

编译路径第一次被 SD 场景覆盖。对团队：Intel (bigPYJ1151) 与 AMD (ganeshr10) 跨厂商协作完成，形成了一套可复用范式——先实测哪些单测在 CPU 可跑、再补一个不空转的 engine 级 e2e、先软失败观察，后续其他平台或新 speculator 方法可直接套用。

- 风险标记：CI 步骤软失败不阻塞合入，硬编码模型与接受长度阈值，依赖多个外部 HF checkpoint，离线回退匹配过宽，内部接口断言耦合，CPU CI 资源占用

关联脉络

- PR #32662 CPU speculative decoding 支持 (PR 正文引用，原标题未在上下文中提供)：本 PR body 明确说明 CPU SD 由 #32662 落地，引入 `csrc/cpu/spec_decode_utils.cpp` 与 `vllm/utils/cpu_triton_utils.py`，而这些文件此前在任何平台都没有测试引用；本 PR 正是为它补上 CI 覆盖与功能矩阵声明。
- PR #56763 [CI] Update entrypoints CI: 同属 `.buildkite` 下 CI 步骤的编排与拆分调整 (该 PR 按 API 目录重命名并重分 entrypoints 用例)，与本 PR 新增 CPU 测试区域属同一 CI 演进脉络，可对照 CI 步骤粒度与触发条件的取舍。