PR #51905 完整报告

vllm-project/vllm

[XPU][CI]Change to use global VLLM_DISABLE_COMPILE_CACHE=1 in Intel GPU CI

合并时间：2026-08-12 12:30

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/51905

执行摘要

- 一句话：Intel GPU CI 全局改用 VLLM_DISABLE_COMPILE_CACHE=1
- 推荐动作：可作为 CI 配置收口的小型示例快速浏览，无需精读源码。值得关注的设计决策是：将跨多个 YAML 重复的环境变量前置改为基础脚本全局导出，并用 docker run -e 透传保证容器内外一致，这种模式适用于同类硬件 CI 的批量配置收敛。

功能与动机

PR body 明确提出 "Change to use global VLLM_DISABLE_COMPILE_CACHE=1 in Intel GPU CI"。此前需要禁用编译缓存的测试任务都要在 pytest 命令前手动加 VLLM_DISABLE_COMPILE_CACHE=1，并在多个 YAML 中反复维护 "# Remove VLLM_DISABLE_COMPILE_CACHE=1 once intel/intel-xpu-backend-for-triton#7682 is fixed" 注释，容易遗漏且难以维护；统一为脚本级全局导出后，新增测试任务不再需要关心该变量。

实现拆解

1. 脚本层收口：在 .buildkite/scripts/hardware_ci/run-intel-test.sh 中，于导出 PYTHONPATH 之后新增 export VLLM_DISABLE_COMPILE_CACHE=1，并将 VLLM_DISABLE_COMPILE_CACHE 追加到待转发环境变量列表（HF_TOKEN、ZE_AFFINITY_MASK 等）中，同时在 docker run 命令行补充 -e VLLM_DISABLE_COMPILE_CACHE 参数，保证容器内外测试环境一致。
2. 配置层清理：.buildkite/intel_jobs 下的 7 个 YAML 文件（model_executor_intel.yaml、model_runner_v2_intel.yaml、models_distributed_intel.yaml、basic_correctness.yaml、benchmarks_intel.yaml、samplers_intel.yaml、test-intel.yaml）统一删除 commands 中的 VLLM_DISABLE_COMPILE_CACHE=1 前缀以及对应的上游修复注释；.buildkite/scripts/hardware_ci/run-intel-ci-test.sh 也同步清理了 v1 分支命令中的同类前缀与注释。
3. 验证方式：维护者通过 /ci run 触发 Buildkite CI（#83499，commit ab58708a9190）做全量 Intel GPU 验证；本次变更无源码或测试配套改动。

关键文件：

- .buildkite/scripts/hardware_ci/run-intel-test.sh（模块 CI 脚本；类别 infra；类型 infrastructure）：核心改动文件：在脚本顶部全局导出

VLLM_DISABLE_COMPILE_CACHE=1，并加入 docker run 的 -e 透传参数，是本次配置收口的真正落点。

- .buildkite/scripts/hardware_ci/run-intel-ci-test.sh（模块 CI 脚本；类别 infra；类型 infrastructure）：同步清理 v1 分支测试命令中的 VLLM_DISABLE_COMPILE_CACHE=1 前缀与修复注释，保持与全局导出一致。
- .buildkite/intel_jobs/test-intel.yaml（模块 CI 任务；类别 config；类型 configuration）：汇总多个 V1 测试步骤的 Intel CI 任务文件，多处 pytest 命令被清理。
- .buildkite/intel_jobs/model_executor_intel.yaml（模块 CI 任务；类别 config；类型 configuration）：Model Executor Intel 测试任务的代表样例，清理后展示配置收口效果。
- .buildkite/intel_jobs/model_runner_v2_intel.yaml（模块 CI 任务；类别 config；类型 configuration）：V2 Model Runner 的 Intel 测试任务，多条命令集中清理变量前缀与注释。
- .buildkite/intel_jobs/models_distributed_intel.yaml（模块 CI 任务；类别 config；类型 configuration）：分布式模型测试任务，test_sharded_state_loader 命令简化。
- .buildkite/intel_jobs/basic_correctness.yaml（模块 CI 任务；类别 config；类型 configuration）：基础正确性测试任务，test_cpu_offload.py 命令简化。
- .buildkite/intel_jobs/benchmarks_intel.yaml（模块 CI 任务；类别 config；类型 configuration）：Benchmarks Intel 测试任务，benchmarks/ 命令简化。
- .buildkite/intel_jobs/samplers_intel.yaml（模块 CI 任务；类别 config；类型 configuration）：Sampler Intel 测试任务，保留 VLLM_XPU_USE_SAMPLER_KERNEL=1 但移除编译缓存变量。

关键符号：未识别

关键源码片段

.buildkite/intel_jobs/model_executor_intel.yaml

Model Executor Intel 测试任务的代表样例，清理后展示配置收口效果。

```
# 整理后关键片段：VLLM_DISABLE_COMPILE_CACHE=1 已由 run-intel-test.sh 全局导出，
# 这里不再需要逐条前置变量，也不再保留修复注释。
commands:
  - >-
    bash .buildkite/scripts/hardware_ci/run-intel-test.sh
    'apt-get update && apt-get install -y curl libsodium23 &&
    pip3 install tensorizer==2.10.1 &&
    pip3 install runai-model-streamer[s3,gcs,azure]\>=0.15.7 &&
    export VLLM_WORKER_MULTIPROC_METHOD=spawn &&
    export PYTHONFAULTHANDLER=1 &&
    cd tests &&
    pytest -v -s model_executor -m "not slow_test" --ignore="model_executor/layers/test_rocm_
    unquantized_gemm.py" --deselect="tests/model_executor/model_loader/test_reload.py::test_kv_
    scale_reload"'
```

评论区精华

review 中无实质技术争论：claude[bot] 提示该 PR 来自 fork，默认禁用自动 review，维护者可评论 @claude review 手动触发；维护者 jikunshang 直接发出 /ci run 触发 Buildkite CI #83499 后批准合并。整个过程符合纯 CI 配置收口的预期。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 全局导出会让所有走 run-intel-test.sh 的 Intel GPU 测试任务继承 VLLM_DISABLE_COMPILE_CACHE=1，当前是预期行为，但若未来某个任务需要启用编译缓存，必须把全局变量改为更细粒度的任务级控制，否则删除全局导出会影响所有任务。
  2. 该变量本质是绕过上游 intel/intel-xpu-backend-for-triton#7682 的已知问题，脚本内仍保留修复注释；上游修复后需人工跟进清理，长期带着 workaround 存在被遗忘的风险。
  3. 影响面明确限定在 .buildkite/ 下的 Intel GPU CI 配置与脚本，不涉及 vLLM 运行时、模型逻辑或用户路径，回归风险很低。 - 影响：对 CI 维护者影响最大：新增或调整 Intel GPU 测试任务时不再需要记住是否要加 VLLM_DISABLE_COMPILE_CACHE=1，配置更简洁一致；对 CI 实际行为无变化，因为所有命令之前已经显式设置了该变量，本次只是把分散设置改为脚本级统一设置；对 vLLM 用户和运行时无影响。 - 风险标记：环境变量全局生效，依赖上游 #7682 修复，CI 配置集中化

关联脉络

- PR #51759 [CI/Release][1/N][XPU] Publish XPU Triton shim index: 同为 Intel GPU/XPU CI 基础设施演进，涉及 XPU 上 Triton 的构建与发布链路，与本 PR 的 XPU 编译缓存禁用逻辑同属一条技术线。
- PR #51877 [ROCm][CI] Speed Up ROCm Skinny GEMM Tests (reduced parameterizations, : 同为硬件特定 CI 的收敛与提速方向，可对比跨平台 CI 配置的维护模式。