

PR #56819 完整报告

vllm-project/vllm

[CI] Pass the scale-out endpoint flag in EC E2E

合并时间: 2026-09-15 12:25

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/56819>

执行摘要

- 一句话: 修复 EC E2E 脚本未同步 `--enable-scale-out` 导致的 CI 404
- 推荐动作:

功能与动机

PR #55176 合并时把环境变量 `gate` 替换为 `vllm serve --enable-scale-out` 参数, 但 L4 Scale-out EC E2E 脚本仍引用被删除的环境变量。PR body 明确指出: Main build #88790 因此 "started healthy encode and prefill servers without the generate route", 且原任务与自动重试都以 HTTP 404 失败; 而 "The exact Scale-out EC job did not run in PR #55176's green build #88768", 导致回归未被拦截。khluu 在 issue 评论中进一步证实这是确定性 main 分支破坏: main #88913 原任务、自动重试以及 daily-main #88920 均复现相同 404 签名, 而非偶发 runner 故障。

实现拆解

实现分三步完成, 均为测试脚本与文档的同步修复, 不涉及产品代码:

1. 修改 E2E 脚本 `encode` 实例启动命令: 在 `tests/entrypoints/scale_out/ec_integration/run_scale_out_ec_e2e_test.sh` 的 `run_scale_out_ec()` 函数中, 将 `env CUDA_VISIBLE_DEVICES="$GPU_E" VLLM_ENABLE_SCALE_OUT_ENDPOINTS=1` 改为 `env CUDA_VISIBLE_DEVICES="$GPU_E"`, 并在 `vllm serve "$MODEL"` 之后新增 `--enable-scale-out` 参数。原因是该实例作为 EC producer 需要暴露 `/inference/v1/generate` 端点供 E2E 验证请求调用。
2. 同步修改 `prefill` 实例启动命令: 对 `GPU_PD` 上的 `prefill/decode` EC consumer 实例做完全相同的替换。该实例同样需要 `scale-out` 端点处理 `encode` 与生成请求, 此前因缺少路由导致作业在首个 `encode` 请求即 404。
3. 更新安全文档: `docs/usage/security.md` 中 `/v1/responses/render` 端点的可用条件从 `"VLLM_ENABLE_SCALE_OUT_ENDPOINTS=1"` 改为 `"--enable-scale-out is set"`, 与新的 CLI 开关保持一致, 避免文档继续误导用户使用已移除的环境变量。
4. 验证配套: 作者在本地执行了 `bash -n` 语法检查、`shellcheck`、`pre-commit run`、`git diff --check`, 并做了静态断言 (确认脚本中恰好存在两个 `--enable-scale-out` 参数且无旧环境变量残留); 随后在 exact L4 Scale-out EC Connector E2E lane 上跑通 (Build #88826, 退出码 0, 3m45s), 单图、双图与纯文本输出均与 baseline 一致。

关键文件：

- tests/entrypoints/scale_out/ec_integration/run_scale_out_ec_e2e_test.sh（模块 E2E 测试；类别 test；类型 test-coverage；符号 run_scale_out_ec）：这是本次回归的源头文件：仍使用已移除的 VLLM_ENABLE_SCALE_OUT_ENDPOINTS 环境变量，导致 encode/prefill 服务不注册 /inference/v1/generate 路由。修复后脚本恢复对 scale-out E2E 路径的真实拦截能力。
- docs/usage/security.md（模块 使用文档；类别 docs；类型 documentation）：同步更新 /v1/responses/render 端点的启用方式说明，避免文档继续引导用户使用已移除的环境变量。

关键符号：run_scale_out_ec

关键源码片段

tests/entrypoints/scale_out/ec_integration/run_scale_out_ec_e2e_test.sh

这是本次回归的源头文件：仍使用已移除的 VLLM_ENABLE_SCALE_OUT_ENDPOINTS 环境变量，导致 encode/prefill 服务不注册 /inference/v1/generate 路由。修复后脚本恢复对 scale-out E2E 路径的真实拦截能力。

```
# 修正后的 run_scale_out_ec() 核心实例启动部分。
# 背景：PR #55176 将 scale-out 端点开关从环境变量 VLLM_ENABLE_SCALE_OUT_ENDPOINTS
# 迁移为 vllm serve 的 --enable-scale-out 参数。旧脚本残留环境变量，
# 导致 encode/prefill 服务启动时不注册 /inference/v1/generate 路由，
# 首个 encode 请求确定性返回 HTTP 404。
run_scale_out_ec() {
    # ...
    # Encode-only EC producer 实例。它运行视觉编码器并通过
    # ECExampleConnector 共享存储发布 embeddings。
    # 必须显式传 --enable-scale-out，否则生成路由不注册。
    echo "Starting encode instance on GPU $GPU_E, port $ENCODE_PORT"
    env CUDA_VISIBLE_DEVICES="$GPU_E" \
        vllm serve "$MODEL" \
        --enable-scale-out \
        --port "$ENCODE_PORT" \
        --max-model-len "$MAX_MODEL_LEN" \
        --enforce-eager \
        --gpu-memory-utilization 0.01 \
        --no-enable-prefix-caching \
        --max-num-batched-tokens 114688 \
        --max-num-seqs "$MAX_NUM_SEQS" \
        --ec-transfer-config '{
            "ec_connector": "ECExampleConnector",
            "ec_role": "ec_producer",
            "ec_connector_extra_config": {
                "shared_storage_path": "'"$EC_SHARED_STORAGE_PATH"'"
            }
        }' \
        > "$ENCODE_LOG" 2>&1 &
    PIDS+=("$!")
}
```

```

# Prefill/decode EC consumer 实例。它接收 metadata-only 特征加
# ec_transfer_params, 并加载 encode 实例发布的 embeddings。
# 与 encode 实例一样需要 --enable-scale-out 以暴露 scale-out 端点。
echo "Starting prefill instance on GPU $GPU_PD, port $PREFILL_PORT"
env CUDA_VISIBLE_DEVICES="$GPU_PD" \
    vllm serve "$MODEL" \
    --enable-scale-out \
    --port "$PREFILL_PORT" \
    --max-model-len "$MAX_MODEL_LEN" \
    --enforce-eager \
    --enable-mm-embeds \
    --gpu-memory-utilization "$GPU_MEMORY_UTILIZATION" \
    --max-num-seqs "$MAX_NUM_SEQS" \
    --ec-transfer-config '{
        "ec_connector": "ECExampleConnector",
        "ec_role": "ec_consumer",
        "ec_connector_extra_config": {
            "shared_storage_path": "'"$EC_SHARED_STORAGE_PATH"'"
        }
    }' \
    > "$PREFILL_LOG" 2>&1 &
PIDS+=("$!")
}

```

评论区精华

该 PR 没有 reviewer 评论 (review_comments_count 为 0)，核心讨论集中在 issue 评论与 PR body 中：

- exact 硬件验证证据：khluu 在 issue 评论中给出 Build #88826 通过、exact lane 退出码 0 的证据，并说明 baseline 与 scale-out 服务均成功启动，单图 / 双图 / 纯文本输出与 baseline 一致。
- main 分支确定性失败证据：khluu 补充说明 main #88913 原任务与自动重试均以相同 404 签名失败，daily-main #88920 再次复现，并强调 "This is now repeated deterministic main breakage, not a one-off runner failure." 这直接支撑了修复的必要性，也说明 CI 在此前已失去对该路径的拦截能力。
- 重复 PR 排查：作者在 PR body 中说明已查阅 issue #55117 及其评论，并搜索 open PR，确认附近结果均为 Rust 路由支持或无关 scale-out 功能，没有其他 PR 更新该 E2E launcher，排除重复提交。
- AI 辅助声明：作者披露使用 OpenAI Codex 诊断 CI 失败并准备补丁，同时声明 "A human submitter must review every changed line"。
 - exact lane 验证与 main 分支确定性失败证据 (testing): 修复有效，隔离的 exact L4 lane 保持绿色；main 分支回归被坐实并已修复。
 - 重复 PR 排查 (question): 确认非重复提交，可以合入。
 - AI 辅助补丁披露 (other): PR 获得 tlmchlsmith 批准，无反对意见。

风险与影响

- 风险：风险较低，但仍有几个需要留意的点：
 1. 旧环境变量残留未全量排查：本 PR 只修复并断言了 `run_scale_out_ec_e2e_test.sh` 这一个脚本，仓库内其他平台（如 AMD/Intel/ROCm）的 scale-out 或 EC 相关脚本是否仍引用 `VLLM_ENABLE_SCALE_OUT_ENDPOINTS` 并未在 PR 中说明。若存在同类残留，后续会在其他 CI lane 上以相同方式失败。
 2. 健康检查不覆盖路由存在性：`wait_for_server` 只等待端口可连与进程存活，不会校验 `/inference/v1/generate` 路由是否注册。这正是本次回归能静默通过服务启动阶段、直到首次 encode 请求才暴露 404 的原因。未来若 flag 再次丢失，CI 仍会以同样方式延迟失败。
 3. 文档与代码的双点维护：`docs/usage/security.md` 的同步修改依赖人工发现，若后续再发生类似的配置项迁移，文档与测试脚本都有再次脱节的可能。 - 影响：
 - 风险标记：配置迁移未同步测试脚本，健康检查不覆盖路由存在性，旧环境变量可能残留其他脚本

关联脉络

- PR #55176 以 `--enable-scale-out` 替换 scale-out 环境变量 `gate`（标题据 PR body 表述还原）：该 PR 完成配置迁移但未同步更新 L4 Scale-out EC E2E 脚本，是本 PR 所修复回归的直接源头。
- PR #55117 scale-out 相关 issue（作者 duplicate check 引用）：作者在 PR body 中说明已查阅该 issue 及其评论，用于排除重复 PR 的可能。