PR #42310 完整报告

vllm-project/vllm

[CI][PD] Add optional/nightly DSv4 Disaggregated eval

合并时间：2026-07-16 05:04

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/42310

执行摘要

- 一句话：新增 DSv4-Flash 可选精度测试并修复 GPU 分配 bug
- 推荐动作：该 PR 主要是 CI/ 测试改进，适合 CI 维护者关注。值得借鉴的是对 Bash 算术陷阱的修复，可作为多 GPU 测试脚本编写的注意事项。新增的可选测试为重要模型提供了回归保护。

功能与动机

PR body 原文：'As per-title, add DSv4 8 gpus tests. Optional as we don't want to run this gpu-hungry setup on each PR.'

实现拆解

1. CI 配置新增步骤：在 .buildkite/test_areas/disaggregated.yaml 中添加名为 DSv4-Flash Disaggregated DP EP 的步骤，指定 8 个 H200 GPU，设置环境变量（PREFILLER_TP_SIZE=4、DECODER_TP_SIZE=4、DP_EP=1 等），并标记为 optional: true 和超时 60 分钟。该步骤通过 install-kv-connectors.sh 安装依赖后执行 run_accuracy_test.sh。
2. 修复 GPU 分配算法：在 tests/v1/kv_connector/nixl_integration/run_accuracy_test.sh 中，原有逻辑在循环中复用 GPU_ID 变量进行算术运算，当 GPU_ID 变为逗号分隔的字符串后，Bash 的 ((GPU_ID + j)) 会将其解释为逗号运算符，导致 GPU 分配重叠。修复引入了 GPU_START 和 DECODE_START 局部变量，确保模运算基于起始索引，从而正确分配 GPU 资源。
3. 补充预期准确率：在 tests/v1/kv_connector/nixl_integration/test_accuracy.py 的 EXPECTED_VALUES 字典中添加 'deepseek-ai/DeepSeek-V4-Flash': 0.95，用于端到端精度验证。

关键文件：

- tests/v1/kv_connector/nixl_integration/run_accuracy_test.sh（模块 测试脚本；类别 test；类型 test-coverage）：修复了 GPU 分配的关键 bug，该 bug 可能影响所有使用该脚本的多 TP 测试，确保 prefill 和 decode 实例分配到不同 GPU。
- .buildkite/test_areas/disaggregated.yaml（模块 CI 配置；类别 config；类型 configuration）：新增 DSv4-Flash 的可选测试步骤，是 PR 的主要 CI 配置变更。

- tests/v1/kv_connector/nixl_integration/test_accuracy.py（模块 精度测试；类别 test；类型 test-coverage）：为 DSv4-Flash 模型添加预期准确率阈值，是测试验证的必要条件。

关键符号：run_tests_for_model, test_accuracy

关键源码片段

tests/v1/kv_connector/nixl_integration/run_accuracy_test.sh

修复了 GPU 分配的关键 bug，该 bug 可能影响所有使用该脚本的多 TP 测试，确保 prefill 和 decode 实例分配到不同 GPU。

```
# prefill 实例 GPU 分配循环（run_tests_for_model 函数内部）
for i in $(seq 0 $((NUM_PREFILL_INSTANCES-1))); do
    # 计算起始 GPU 索引，避免在使用 GPU_ID 字符串做算术时被 Bash 解释为逗号运算符
    GPU_START=$((i % $(get_num_gpus)))
    GPU_ID=$GPU_START
    NEXT_GPU=$GPU_START
    PREFILLER_WORLD_SIZE=$((PREFILLER_TP_SIZE * PREFILLER_PP_SIZE))
    for (( j=1; j < PREFILLER_WORLD_SIZE; j++ )); do
        # 使用 GPU_START 而非 GPU_ID 进行模运算，确保得到正确的下一个 GPU 索引
        NEXT_GPU=$(((GPU_START + j) % $(get_num_gpus)))
        GPU_ID=${GPU_ID},${NEXT_GPU}
    done
    # ... 使用 GPU_ID 启动 prefill 实例 ...
done

# decode 实例 GPU 分配循环
for i in $(seq 0 $((NUM_DECODE_INSTANCES-1))); do
    # 类似地，使用 DECODE_START 避免逗号运算符问题
    DECODE_START=$(((i + NEXT_GPU + 1) % $(get_num_gpus)))
    GPU_ID=$DECODE_START
    NEXT_GPU=$DECODE_START
    for (( j=1; j < DECODER_TP_SIZE; j++ )); do
        NEXT_GPU=$(((DECODE_START + j) % $(get_num_gpus)))
        GPU_ID=${GPU_ID},${NEXT_GPU}
    done
    # ... 使用 GPU_ID 启动 decode 实例 ...
done
```

评论区精华

评论重点

- gemini-code-assist[bot]指出新增步骤缺少 source_file_dependencies 块，与其他步骤不一致，建议添加以正确管理缓存。但合并版本未包含该块，可能因 optional 步骤不需要精确依赖跟踪。
- claude[bot]发现 run_accuracy_test.sh 中的 GPU 分配 bug：当 GPU_ID 变为逗号字符串后，Bash 算术运算会触发逗号运算符，导致 GPU 冲突。作者在后续提交中修复。

- AndreasKaratzas批准 PR，并建议后续在 AMD 上镜像此测试。
- 新 CI 步骤缺少 source_file_dependencies (design): 最终合并版本未包含此块，可能因 optional 步骤不需要精确缓存依赖，或 reviewer 未强求。
- GPU 分配 Bash 算术 bug (correctness): 作者在提交 fix gpu assignment 中引入 GPU_START/DECODE_START 变量修复此问题。

风险与影响

- 风险：新增步骤使用 8 个 H200 GPU，可能加剧 CI 资源争用，但标记为 optional 不阻塞门禁。GPU 分配 bug 已修复，不影响现有测试。测试依赖 kv-connector 环境，若安装失败会超时退出。总体风险低。
- 影响：直接影响：仅当手动或夜间运行时占用 8 个 H200 GPU 几分钟。对用户无影响。为 DSv4-Flash 模型提供 Disaggregated 精度基线，便于回归检测。CI 配置增加 22 行，测试脚本修改 13 行。
- 风险标记：可选测试 GPU 消耗高，Bash 算术 bug（已修复），仅影响可选测试

关联脉络

- 暂无明显关联 PR