

PR #40990 完整报告

vllm-project/vllm

[ROCm][CI] Extend ROCm quick reduce coverage

合并时间: 2026-05-26 21:57

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/40990>

执行摘要

- 一句话: 扩展 ROCm 快速归约测试覆盖与 CI 集成
- 推荐动作: 此 PR 整体健康, 具有正面价值, 建议合并。考虑采纳 review 建议, 改进断言的严谨性和错误处理, 以提升测试的可靠性。值得 ROCm 相关开发者精读, 特别是学习如何编写分布式进程测试、利用 Ray 进行多 GPU 测试以及配置 CI 依赖声明。

功能与动机

ROCm 平台的 quick-reduce 功能缺乏专项测试。PR body 说明: 'The kernel-level quick-reduce behavior now lives in `test_quick_all_reduce.py`, while the ROCm-specific file keeps the env parsing, communicator policy, and model-level end-to-end checks.' 需集中 kernel 级测试并补充 ROCm 特定的单元与端到端测试。

实现拆解

1. 新建 ROCm 专属测试文件: `tests/distributed/test_rocm_quick_reduce.py` 包含辅助函数 `_make_quick_allreduce` 构造模拟对象、`_quick_allreduce_worker` 在独立进程中执行 quick-reduce 并验证结果、`_run_two_gpu_quick_allreduce_test` 管理多进程启动, 以及 `_build_prompt`、`_log_prompt_summaries`、`_get_model_path` 等端到端生成测试所需工具。该文件受 `pytest.mark.skipif` 保护, 仅当平台为 ROCm 时才执行。
2. 增强通用 quick-reduce 测试: `tests/distributed/test_quick_all_reduce.py` 新增 `on_gfx942` 检测函数以跳过已知的 CUDA graph 捕捉问题, 提取 `_assert_quickreduce` 公用断言, 并引入 `bf16_cast_quickreduce` 远程测试函数专门验证 BF16 输入经过 cast 后的正确性。同时, 在 `graph_quickreduce` 和 `eager_quickreduce` 中添加了对 `HIP_VISIBLE_DEVICES` 和 `ROCR_VISIBLE_DEVICES` 的清理, 避免多 GPU 环境下的设备冲突。
3. 重构 Ray 并行测试辅助: `tests/utils.py` 中的 `multi_process_parallel` 函数, 将原先的 `working_dir` 上传方式改为通过 `PYTHONPATH` 环境变量传递仓库路径, 并利用 `current_platform.ray_noset_device_env_vars` 设置设备环境变量。这一改动规避了上传完整源码树时 Ray 对包大小的限制, 同时保留调试便利性。
4. 更新 CI 配置: `.buildkite/test-amd.yaml` 为 MI355 节点的 Distributed Tests 步骤增加了大量源文件依赖 (包括 `csrc/custom_quickreduce.cu`、`vllm/envs.py` 等), 并将新增的 `test_rocm_quick_reduce.py` 和 `test_quick_all_reduce.py` 加入测试命令, 确保 CI 会自动运行这些测试。

关键文件：

- tests/distributed/test_rocm_quick_reduce.py (模块 ROCm 测试；类别 test；类型 test-coverage；符号 _log, _reload_envs, _make_quick_allreduce, _quick_allreduce_worker)：新增文件，包含 ROCm 专属的 quick-reduce 单元测试、环境解析、多进程验证和模型端到端测试，是整个 PR 的核心。
- tests/distributed/test_quick_all_reduce.py (模块 快速归约测试；类别 test；类型 test-coverage；符号 on_gfx942, _assert_quickreduce, bf16_cast_quickreduce, test_custom_quick_allreduce_bf16_cast)：添加了 on_gfx942 检测、_assert_quickreduce 辅助函数和新的 bf16 cast 测试，增强了通用 quick-reduce 正确性验证。
- tests/utlis.py (模块 测试工具；类别 test；类型 test-coverage)：修改 multi_process_parallel 实现，改用 PYTHONPATH 环境变量传递仓库路径，避免 Ray 上传限制，并清理了设备环境变量设置。
- .buildkite/test-amd.yaml (模块 CI 配置；类别 config；类型 configuration)：为 MI355 分布式测试步骤添加 quick-reduce 相关的源文件依赖和测试命令，确保新测试在 CI 中自动执行。

关键符号：_log, _reload_envs, _make_quick_allreduce, _quick_allreduce_worker, _run_two_gpu_quick_allreduce_test, _build_prompt, _log_prompt_summaries, _get_model_path, on_gfx942, _assert_quickreduce, bf16_cast_quickreduce, test_custom_quick_allreduce_bf16_cast, multi_process_parallel

关键源码片段

tests/distributed/test_rocm_quick_reduce.py

新增文件，包含 ROCm 专属的 quick-reduce 单元测试、环境解析、多进程验证和模型端到端测试，是整个 PR 的核心。

```
def _quick_allreduce_worker(
    rank: int,
    port: int,
    quant_level: str,
    dtype_name: str,
    cast_bf16: bool,
):
    # 设置 ROCm 环境变量以便 QuickAllReduce 使用指定的量化级别与 BF16 转换
    os.environ["VLLM_ROCM_QUICK_REDUCE_QUANTIZATION"] = quant_level
    os.environ["VLLM_ROCM_QUICK_REDUCE_CAST_BF16_TO_FP16"] = "1" if cast_bf16 else "0"
    _log(
        f"worker start: rank={rank} quant={quant_level} "
        f"dtype={dtype_name} cast_bf16={cast_bf16}"
    )

    device = torch.device(f"cuda:{rank}")
    torch.accelerator.set_device_index(device)
    # 使用 gloo 后端初始化进程组，避免 NCCL 在多进程下的潜在问题
```

```

dist.init_process_group(
    backend="gloo",
    init_method=f"tcp://127.0.0.1:{port}",
    rank=rank,
    world_size=WORLD_SIZE,
)

qar = None
try:
    from vllm.distributed.device_communicators.quick_all_reduce import (
        QuickAllReduce,
    )

    qar = QuickAllReduce(group=dist.GroupMember.WORLD, device=rank)
    assert not qar.disabled

    # 根据数据类型确定元素数量: float16 使用 8MB, 其他使用 4MB
    num_elements = 8 * MB if dtype_name == "float16" else 4 * MB

    dtype = getattr(torch, dtype_name)
    inp = torch.ones(num_elements, dtype=dtype, device=device)

    # 断言应触发 quick-allreduce 而非回退
    assert qar.should_quick_allreduce(inp)
    if cast_bf16:
        # 若开启 cast, 验证内部使用了 fp16 内核
        assert qar.use_fp16_kernels

    out = qar.quick_all_reduce(inp)
    # 验证结果近似等于 2 倍输入 (WORLD_SIZE=2), 容差较宽松
    assert torch.allclose(out, inp * WORLD_SIZE, atol=2.5, rtol=0.1)
    _log(
        f"worker complete: rank={rank} quant={quant_level} "
        f"dtype={dtype_name} num_elements={num_elements} "
        f"use_fp16_kernels={qar.use_fp16_kernels}"
    )
finally:
    if qar is not None:
        qar.close()
    if dist.is_initialized():
        dist.destroy_process_group()

```

tests/distributed/test_quick_all_reduce.py

添加了 on_gfx942 检测、_assert_quickreduce 辅助函数和新的 bf16 cast 测试, 增强了通用 quick-reduce 正确性验证。

```

def on_gfx942() -> bool:
    # 仅在 ROCm 平台时检查是否为 gfx942 架构
    if current_platform.is_rocm():

```

```
from vllm.platforms.rocm import on_gfx942 as rocm_on_gfx942
return rocm_on_gfx942()
return False
```

```
def _assert_quickreduce(fa, inp):
    # 验证 quick-reduce 通信器已创建、未被禁用且输入尺寸应触发 quick-allreduce
    assert fa is not None
    assert not fa.disabled
    assert fa.should_quick_allreduce(inp)
```

评论区精华

- 正确性检查过于宽松: gemini-code-assist[bot] 指出 `_quick_allreduce_worker` 使用 `torch.ones` 和 `atol=2.5` 的断言几乎无法捕捉错误, 建议改用随机输入并收紧容差。同样, `_check_tp_allreduce_uses_quick_reduce` 中也有类似问题。作者未在评论中直接回应, 但 PR 最终通过了审核。
- 缺少错误处理: 同一助手建议在 `_quick_allreduce_worker` 中添加 `try-except` 并用 `traceback.format_exc()` 记录异常以便调试子进程崩溃。该建议未在后续讨论中解决。
- 平台特定导入需加保护: 审阅者 `tjtanaa` 要求避免直接导入 `vllm.platforms.rocm.on_gfx942` 而应检查 `current_platform.is_rocm()`。作者 `AndreasKaratzas` 确认修复并同时修正了其他类似未保护导入。
- gfx942 CUDA graph 问题: `AndreasKaratzas` 注释提到在 gfx942 上 CUDA graph 捕获与 quick-reduce 配合时遇到 `hipErrorStreamCaptureInvalidated`, 因此将 `graph_quickreduce` 标记为 `xfail`, 并建议合并后归档 issue。
 - 测试断言容差过松 (correctness): 未被直接采纳, PR 已合并但该建议未解决。
 - 子进程缺少错误处理 (testing): 未被采纳, 代码中未添加错误处理。
 - 平台特定导入需保护 (design): 作者确认修复, 并改正了其他类似未保护导入。
 - gfx942 CUDA graph 捕获问题 (question): 已被标记为 `xfail`, 待后续修复。

风险与影响

- 风险:
 - 回归风险: 本 PR 只涉及测试和 CI 配置, 无核心代码变更, 回归风险极低。但新增的测试可能因环境或时序而失败, 需要持续维护。
 - 测试有效风险: 如 review 所指, 部分断言容差过大 (如 `atol=2.5`), 可能导致对潜在 bug 的漏检测。建议后续加强断言, 使用随机输入或更紧的容差。
 - CI 时长增加: 新增两个测试文件, 且在 MI355 节点上运行, 会延长 CI 执行时间。但 quick-reduce 测试执行较快, 影响有限。
 - 平台隔离: `on_gfx942` 检测仅在 ROCm 下有效, 已通过导入保护避免跨平台崩溃。
- 影响:
 - 用户影响: 无直接影响, 因仅改动测试和 CI。

- 系统影响：提升了 vLLM 在 ROCm 平台上的测试覆盖度，特别是 QuickAllReduce 模块。有助于在开发过程中及早发现归约通信错误，对 MI300/350 等 AMD GPU 用户是正面改进。
- 团队影响：测试维护工作量略有增加，但受益于更可靠的回归检测。CI 配置的调整使后续扩展测试更加规范。
- 风险标记：断言容差宽松，子进程错误处理不足，CI 时长增加

关联脉络

- PR #43629 [ROCm] Remove MegaMoE integration in deepseek v4: 同为 ROCm 平台维护工作，清理 AMD 特定代码。
- PR #43303 [Misc][Refactor][ROCm] Convert MoRI-related envvars to extra config args: 同为 ROCm 相关重构，涉及环境变量迁移。