PR #48843 完整报告

vllm-project/vllm

[BugFix] Set graph_pool_id before FULL CUDA graph capture in ModelRunner V2

合并时间：2026-07-21 20:59

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/48843

执行摘要

- 一句话：修复 FULL CUDA 图捕获前未设置 graph_pool_id 导致断言失败
- 推荐动作：PR 修复了关键 bug，逻辑正确且附带单元测试。建议合并，但需确保测试 CI 接入。代码简洁，值得精读以理解 CUDA 图捕获的配套设置。

功能与动机

Issue #48840 报告在使用 VLLM_USE_NCCL_SYMM_MEM=1 和 ModelRunner V2 FULL CUDA 图时，断言失败：graph_pool_id is not set under graph capture。需要确保进入 torch.cuda.graph() 前设置 graph_pool_id 以兼容 NCCL 对称内存。

实现拆解

1. 在 vllm/v1/worker/gpu/cudagraph_utils.py 导入 set_graph_pool_id 函数。
2. 在 CudaGraphManager.capture() 的 FULL 分支中、torch.cuda.graph() 之前，根据 self.pool 是否为空：若非空则 set_graph_pool_id(self.pool)，否则 set_graph_pool_id(current_platform.graph_pool_handle())。
3. 新增 tests/v1/cudagraph/test_cudagraph_manager.py 测试文件，包含 _reset_graph_pool_id fixture、_create_vllm_config 辅助函数和 test_full_capture_sets_graph_pool_id_before_cuda_graph 测试用例，通过 mock 和断言验证捕获前 graph_pool_id 正确设置。

关键文件：

- vllm/v1/worker/gpu/cudagraph_utils.py（模块 CUDA 图；类别 source；类型 core-logic；符号 capture, set_graph_pool_id）：核心修复文件：导入 set_graph_pool_id 并在 FULL CUDA 图捕获路径中调用，确保进入 torch.cuda.graph() 前全局变量已设置。
- tests/v1/cudagraph/test_cudagraph_manager.py（模块 CUDA 图；类别 test；类型 test-coverage；符号 _reset_graph_pool_id, _create_vllm_config, test_full_capture_sets_graph_pool_id_before_cuda_graph, create_forward_fn）：新增测试文件，覆盖 FULL 捕获路径的 graph_pool_id 设置，通过 mock 和断言验证修复正确性。

关键符号：CudaGraphManager.capture, set_graph_pool_id, test_full_capture_sets_graph_pool_id_before_cuda_graph

关键源码片段

vllm/v1/worker/gpu/cudagraph_utils.py

核心修复文件：导入 set_graph_pool_id 并在 FULL CUDA 图捕获路径中调用，确保进入 torch.cuda.graph() 前全局变量已设置。

```
# vllm/v1/worker/gpu/cudagraph_utils.py (head)
from vllm.distributed.device_communicators.pynccl_allocator import set_graph_pool_id

def capture(self, create_forward_fn):
    # ...
    for desc in descs:
        if desc.cg_mode == CUDAGraphMode.FULL:
            graph = torch.cuda.CUDAGraph()
            get_offloader().sync_prev_onload()
            # 设置 graph_pool_id，NCCL 对称内存需要此全局变量
            if self.pool is not None:
                set_graph_pool_id(self.pool)
            else:
                set_graph_pool_id(current_platform.graph_pool_handle())
            with torch.cuda.graph(graph, self.pool):
                forward_fn(CUDAGraphMode.NONE)
                get_offloader().join_after_forward()
            self.graphs[desc] = graph
    # ...
```

tests/v1/cudagraph/test_cudagraph_manager.py

新增测试文件，覆盖 FULL 捕获路径的 graph_pool_id 设置，通过 mock 和断言验证修复正确性。

```
# tests/v1/cudagraph/test_cudagraph_manager.py (head)
@pytest.fixture(autouse=True)
def _reset_graph_pool_id():
    pynccl_allocator._graph_pool_id = None
    yield
    pynccl_allocator._graph_pool_id = None

def test_full_capture_sets_graph_pool_id_before_cuda_graph(monkeypatch):
    graph_pool = object()
    monkeypatch.setattr(gpu_cudagraph_utils, "get_pp_group",
                        lambda: SimpleNamespace(is_first_rank=True, is_last_rank=True))
    monkeypatch.setattr(gpu_cudagraph_utils.current_platform, "get_global_graph_pool",
                        lambda: graph_pool)
    manager = gpu_cudagraph_utils.CudaGraphManager(
        vllm_config=_create_vllm_config(),
        device=torch.device("cpu"),
        cudagraph_mode=CUDAGraphMode.FULL,
        decode_query_len=1,
    )
    desc = BatchExecutionDescriptor(
        cg_mode=CUDAGraphMode.FULL, num_tokens=4, num_reqs=4, uniform_token_count=1)
```

```
    manager._capture_descs[CUDAGraphMode.FULL] = [desc]

    def create_forward_fn(desc, warmup):
        return lambda _mode: None

    # 自定义 torch.cuda.graph 的 __enter__，在其中断言 graph_pool_id 正确
    def cuda_graph_enter(*args, **kwargs):
        assert pynccl_allocator._graph_pool_id is graph_pool

    mock_cuda_graph_ctx = MagicMock()
    mock_cuda_graph_ctx.__enter__ = cuda_graph_enter
    mock_cuda_graph_ctx.__exit__ = MagicMock(return_value=False)

    with (
        patch.object(gpu_cudagraph_utils, "graph_capture", fake_graph_capture),
        patch.object(gpu_cudagraph_utils, "get_offloader", lambda: fake_offloader),
        patch.object(gpu_cudagraph_utils.torch.cuda, "CUDAGraph"),
        patch.object(gpu_cudagraph_utils.torch.cuda, "graph",
                     return_value=mock_cuda_graph_ctx) as mock_cuda_graph,
    ):
        manager.capture(create_forward_fn)
    mock_cuda_graph.assert_called_once()
```

评论区精华

ZJY0516 在 review 中指出新测试 test_cudagraph_manager.py 未在 CI 中运行（标记为 cpu_test 但可能未被触发），未得到回复但 PR 被合并，可能需后续处理。

- 测试未在 CI 中运行 (testing): 未获得回应，PR 被合并，可能需后续处理。

风险与影响

- 风险：变更仅增加 5 行，风险低。但 CUDA 图捕获是性能关键路径，若 set_graph_pool_id 参数错误可能导致图捕获失败或性能问题。测试覆盖了正常路径，未覆盖 pool 为 None 或 graph_pool_handle 失败的 *edge case*。此外，测试未在 CI 中运行，存在回归风险。
- 影响：直接修复在特定配置（VLLM_USE_NCCL_SYMM_MEM=1 + ModelRunner V2 + FULL CUDA 图）下的断言崩溃。影响范围限于使用 NCCL 对称内存的 NVIDIA GPU 用户，该配置常用于大规模推理部署，修复可提升稳定性。
- 风险标记：测试未接入 CI，核心路径变更

关联脉络

- PR #49364 [MRV2] Always build attn metadata at capture time: 同为 CUDA 图捕获改进，涉及 cudagraph_utils 模块。
- PR #49339 [CI] Fix and wire encoder/manager cudagraph unit tests: 同样为 cudagraph 单元测试接入 CI，与本 PR 测试文件相关。

- PR #49302 [Bugfix] Fix DSA crash under breakable piecewise cudagraphs: 另一 CUDA 图相关 bugfix，反映同一模块的持续修复。